



STIC Search Report

EIC 2100

STIC Database Tracking Number: 120959

TO: James D Rutten
Location: 5B46
Art Unit : 2133
Monday, May 03, 2004

Case Serial Number: 09/764526

From: Geoffrey St. Leger
Location: EIC 2100
PK2-4B30
Phone: 308-7800

geoffrey.stleger@uspto.gov

Search Notes

Dear Examiner Rutten,

Attached please find the results of your search request for application 09/764526. I searched Dialog's foreign patent files, technical databases, product announcement files and general files; along with the Internet.

Please let me know if you have any questions.

Regards,

Geoffrey St. Leger
4B30/308-7800

Green, Shirelle

From: Unknown@Unknown.com
Sent: Monday, May 03, 2004 12:16 PM
To: STIC-EIC2100
Subject: Generic form response

FAST & FOCUSED
SEARCH

ResponseHeader=Commercial Database Search Request

AccessDB#= 120959

LogNumber= _____

Searcher= Geoffrey St. Leger

SearcherPhone= 308-7800

SearcherBranch= 4B30/EIC2100

MyDate=Mon May 3 12:16:05 EDT 2004

submitto=STIC-EIC2100@uspto.gov

Name=Derek Rutten

Empno=79877

Phone=703-605-5233

Artunit=2122

Office=pk2-5b46

Serialnum=09/764,526

Earliest=1/17/2001

OtherDate=

Format1=paper

Already1=USP

[Home](#) [Index](#) [Resources](#) [Contact](#) [Internal Search](#)**Scientific and Technical Information Center****[NPL Virtual Library](#) > [Request a Prior Art Search](#)**[Patents Home](#) | [Site Feedback](#)[NPL Virtual Library Home](#) | [STIC Catalog](#) | [Site Guide](#) | [EIC](#) | [Automation Training/ITRPs](#) | [Contact Us](#) | [STIC Staff](#) | [FAQ](#)**STIC EIC2100 Search Request**

Search requests relating to **published applications, patent families, and litigation** may be submitted by filling out this form and clicking on "Send."

For all other search requests, fill out the form, print, and submit the printout with any attachments to the STIC facility serving your Technology Center.

Enter your Contact Information below:

Name:
Employee Number: Phone:
Art Unit or Office: Building & Room Number:

Enter the case serial number (Required):

If not related to a patent application, please enter NA here.

What date would you like to use to limit the search?

Priority Date: Other Date:

Format preferred for results:

☒ Paper ☐ Diskette ☐ E-mail

Where have you searched so far?:

☒ USP ☐ DWPI ☐ EPO ☐ JPO ☐ ACM ☐ IBM TDB ☐ IEEE ☐ INSPEC
☐ SPI ☐ Other

Is this a "Fast & Focused" Search Request? Yes No

A "Fast & Focused" Search is completed in 2-3 hours (maximum). The search must be on a very specific topic and meet certain criteria, which are posted in EIC2100 and on the [NPL Page for TC2100](#).

What is the topic, novelty, motivation, utility, or other specific detail(s) defining the desired focus of this search? Please include the concepts, synonyms, keywords, acronyms, definitions, strategies, and anything else that helps to describe the topic. Please FAX or send the abstract, background, brief summary, pertinent claims, citations of relevant art you have found, or other supplementary information to EIC2100.

For Foreign Patent Family Searches Only
Include the country name and patent number.

Enter your Search Topic Information below:

Looking for a function in source code whose function name includes version information regarding the function. So, for example, if I want to use version 1.2 of the print() function, the name of the function might be print_1.2(). The invention uses this as a dummy function that has no body, but simply uses the name of the function to indicate the version of a library. So the print() function might actually be residing in a library called io.1, and the function would really be called io_1.2(). I already have other art, so I simply need any instance of a function containing version information in the name.

Special Instructions and Other Comments:

(For fastest service, let us know the best times to contact you, in case the searcher needs further clarification on your search.)

I'm usually here from 7-3:30.

Thanks.

Press ALT + F, then P to print this screen for your own information.

SEND **RESET**

USPTO [Intranet Home](#) | [Index](#) | [Resources](#) | [Contacts](#) | [Internet](#) | [Search](#) | [Web Services](#)

Last Modified: 07/24/2003 15:31:52


[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#)^{New!} [more »](#)

library version command

Search

[Advanced Search](#)
[Preferences](#)

Web

Results 1 - 10 of about 801,000 for library version command. (0.63 seconds)

VERSION Command

... The format of the **version** script **commands** is identical ... which symbols are bound to which **version** nodes, and ... not globally visible outside of the shared **library**. ...

www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/gnu-linker/version.html - 14k - [Cached](#) - [Similar pages](#)

Communicator's Command Line Options

... **Command**, Parameters. Create a dummy profile and launch the Account Setup application (through which the user will create a real account). Applies to **version 4.02** ...

developer.netscape.com/docs/manuals/deploymt/options.htm - 14k - [Cached](#) - [Similar pages](#)

HTML Tidy Project Page

... You can find current builds of the **command** line tool, the DLL and the ... All development is now done on the **library version** of Tidy, otherwise known as TidyLib. ...

tidy.sourceforge.net/ - 29k - [Cached](#) - [Similar pages](#)

libwww-perl

... This **version** does not provide HTTP/1.1 support ... Notice, that the **library** is still alpha software (and ... lwp-request A simple **command** that fetch documents specified ...

www.linpro.no/lwp/ - 12k - [Cached](#) - [Similar pages](#)

Visual C++ Developer Center: Microsoft Visual C++ Toolkit 2003

... Toolkit will work fine alongside installed **versions** of Visual ... installs (1) the Visual C++ **command**-line compiler ... 2) the Visual C++ C Runtime **Library** and static ...

msdn.microsoft.com/visualc/vctoolkit2003/ - 23k - May 1, 2004 - [Cached](#) - [Similar pages](#)

Performance Counters Library - command-line options

... **Command**-Line Options -processor -**version** Print information about the processor, **library version**, and rabbit **version**. -**version** subsumes -processor. ...

www.scl.ameslab.gov/Projects/Rabbit/command_version.html - 4k - [Cached](#) - [Similar pages](#)

Ygl Version 3.1 Description

... You can always find the latest **version** of Ygl in ... Several environment variables are referenced in the **library** to control ... x mbx" to the X-Servers **command** line on ...

www.thp.uni-duisburg.de/Ygl/ReadMe.html - 28k - [Cached](#) - [Similar pages](#)

GNU Coding Standards: Command-Line Interfaces

... ways to find out the precise file name where a **command** is found in ... you can do so by printing an additional line of **version** info for each **library** you want ...

www.gnu.org/prep/standards_18.html - 11k - [Cached](#) - [Similar pages](#)

PGPLOT Graphics Subroutine Library

... Starlink also maintains a **version** of PGPLOT layered on the GKS **library** ... and fitting subroutine layered on PGPLOT, and QDP provides a **command** interface to ...

www.astro.caltech.edu/~tjp/pgplot/ - 16k - [Cached](#) - [Similar pages](#)

ASVGuy::SWFBlog: JSFL: Arrange Library

... Comments and bug reports are welcome. Download ASV Arrange **Library command**:

ZIP **version MXP version**. Jan 26, 2004 Update: **Version 1.5** is released. ...

www.asvguy.com/2004/01/jsfl_arrange_li.html - 16k - May 2, 2004 - [Cached](#) - [Similar pages](#)

Client/Server Interface for Berkeley DB

Susan LoVerso
sue@sleepycat.com
Rev 1.3
1999 Nov 29

We provide an interface allowing client/server access to Berkeley DB. Our goal is to provide a client and server library to allow users to separate the functionality of their applications yet still have access to the full benefits of Berkeley DB. The goal is to provide a totally seamless interface with minimal modification to existing applications as well.

The client/server interface for Berkeley DB can be broken up into several layers. At the lowest level there is the transport mechanism to send out the messages over the network. Above that layer is the messaging layer to interpret what comes over the wire, and bundle/unbundle message contents. The next layer is Berkeley DB itself.

The transport layer uses ONC RPC (RFC 1831) and XDR (RFC 1832). We declare our message types and operations supported by our program and the RPC library and utilities pretty much take care of the rest. The *rpcgen* program generates all of the low level code needed. We need to define both sides of the RPC.

DB Modifications

To achieve the goal of a seamless interface, it is necessary to impose a constraint on the application. That constraint is simply that all database access must be done through an open environment. I.e. this model does not support standalone databases. The reason for this constraint is so that we have an environment structure internally to store our connection to the server. Imposing this constraint means that we can provide the seamless interface just by adding a single environment method: DBENV->set_server().

The planned interface for this method is:

```
DBENV->set_server(dbenv,      /* DB_ENV structure */
                 hostname     /* Host of server */
                 cl_timeout,  /* Client timeout (sec) */
                 srv_timeout, /* Server timeout (sec) */
                 flags);      /* Flags: unused */
```

This new method takes the hostname of the server, establishes our connection and an environment on the server. If a server timeout is specified, then we send that to the server as well (and the server may or may not choose to use that value). This timeout is how long the server will allow the environment to remain idle before declaring it dead and releasing resources on the server. The pointer to the connection is stored on the client in the DBENV structure and is used by all other methods to figure out with whom to communicate. If a client timeout is specified, it indicates how long the client is willing to wait for a reply from the server. If the values are 0, then defaults are used. Flags is currently unused, but exists because we always need to have a placeholder for flags and it would be used for specifying authentication desired (were we to provide an authentication scheme at some point) or other uses not thought of yet!

This client code is part of the monolithic DB library. The user accesses the client functions via a new flag to `db_env_create()`. That flag is `DB_CLIENT`. By using this flag the user indicates they want to have the client methods rather than the standard methods for the environment. Also by issuing this flag, the user needs to connect to the server via the `DBENV->set_server()` method.

We need two new fields in the `DB_ENV` structure. One is the socket descriptor to communicate to the server, the other field is the client identifier the server gives to us. The `DB`, and `DBC` only need one additional field, the client identifier. The `DB_TXN` structure does not need modification, we are overloading the `txn_id` field.

Issues

We need to figure out what to do in case of client and server crashes. Both the client library and the server program are stateful. They both consume local resources during the lifetime of the connection. Should one end drop that connection, the other side needs to release those resources.

If the server crashes, then the client will get an error back. I have chosen to implement time-outs on the client side, using a default or allowing the application to specify one through the `DBENV->set_server()` method. Either the current operation will time-out waiting for the reply or the next operation called will time out (or get back some other kind of error regarding the server's non-existence). In any case, if the client application gets back such an error, it should abort any open transactions locally, close any databases, and close its environment. It may then decide to retry to connect to the server periodically or whenever it comes back. If the last operation a client did was a transaction commit that did not return or timed out from the server, the client cannot determine if the transaction was committed or not but must release the local transaction resources. Once the server is back up, recovery must be run on the server. If the transaction commit completed on the server before the crash, then the operation is redone, if the transaction commit did not get to the server, the pieces of the transaction are undone on recover. The client can then re-establish its connection and begin again. This is effectively like beginning over. The client cannot use ID's from its previous connection to the server. However, if recovery is run, then consistency is assured.

If the client crashes, the server needs to somehow figure this out. The server is just sitting there waiting for a request to come in. A server must be able to time-out a client. Similar to `ftpd`, if a connection is idle for `N` seconds, then the server decides the client is dead and releases that client's resources, aborting any open transactions, closing any open databases and environments. The server timing out a client is not a trivial issue however. The generated function for the server just calls `svc_run()`. The server code I write contains procedures to do specific things. We do not have access to the code calling `select()`. Timing out the select is not good enough even if we could do so. We want to time-out idle environments, not simply cause a time-out if the server is idle a while. See the discussion of the server program for a description of how we accomplish this.

Since `rpcgen` generates the `main()` function of the server, I do not yet know how we are going to have the server multi-threaded or multi-process without changing the generated code. The RPC book indicates that the only way to accomplish this is through modifying the generated code in the server. **For the moment we will ignore this issue while we get the core server working, as it is only a performance issue.**

We do not do any security or authentication. Someone could get the code and modify it to spoof messages, trick the server, etc. RPC has some amount of authentication built into it. I haven't yet looked into it much to know if we want to use it or just point a user at it. The changes to the client code

are fairly minor, the changes to our server procs are fairly minor. We would have to add code to a *sed* script or *awk* script to change the generated server code (yet again) in the dispatch routine to perform authentication.

We will need to get an official program number from Sun. We can get this by sending mail to *rpc@sun.com* and presumably at some point they will send us back a program number that we will encode into our XDR description file. Until we release this we can use a program number in the "user defined" number space.

The Server Program

The server is a standalone program that the user builds and runs, probably as a daemon like process. This program is linked against the Berkeley DB library and the RPC library (which is part of the C library on my FreeBSD machine, others may have/need *-lrpc*). The server basically is a slave to the client process. All messages from the client are synchronous and two-way. The server handles messages one at a time, and sends a reply back before getting another message. There are no asynchronous messages generated by the server to the client.

We have made a choice to modify the generated code for the server. The changes will be minimal, generally calling functions we write, that are in other source files. The first change is adding a call to our time-out function as described below. The second change is changing the name of the generated *main()* function to *__dbsrv_main()*, and adding our own *main()* function so that we can parse options, and set up other initialization we require. I have a *sed* script that is run from the distribution scripts that massages the generated code to make these minor changes.

Primarily the code needed for the server is the collection of the specified RPC functions. Each function receives the structure indicated, and our code takes out what it needs and passes the information into DB itself. The server needs to maintain a translation table for identifiers that we pass back to the client for the environment, transaction and database handles.

The table that the server maintains, assuming one client per server process/thread, should contain the handle to the environment, database or transaction, a link to maintain parent/child relationships between transactions, or databases and cursors, this handle's identifier, a type so that we can error if the client passes us a bad id for this call, and a link to this handle's environment entry (for time out/activity purposes). The table contains, in entries used by environments, a time-out value and an activity time stamp. Its use is described below for timing out idle clients.

Here is how we time out clients in the server. We have to modify the generated server code, but only to add one line during the dispatch function to run the time-out function. The call is made right before the return of the dispatch function, after the reply is sent to the client, so that client's aren't kept waiting for server bookkeeping activities. This time-out function then runs every time the server processes a request. In the time-out function we maintain a time-out hint that is the youngest environment to time-out. If the current time is less than the hint we know we do not need to run through the list of open handles. If the hint is expired, then we go through the list of open environment handles, and if they are past their expiration, then we close them and clean up. If they are not, we set up the hint for the next time.

Each entry in the open handle table has a pointer back to its environment's entry. Every operation within this environment can then update the single environment activity record. Every environment can have a

different time-out. The DBENV->set_server call takes a server time-out value. If this value is 0 then a default (currently 5 minutes) is used. This time-out value is only a hint to the server. It may choose to disregard this value or set the time-out based on its own implementation.

For completeness, the flaws of this time-out implementation should be pointed out. First, it is possible that a client could crash with open handles, and no other requests come in to the server. Therefore the time-out function never gets run and those resources are not released (until a request does come in). Similarly, this time-out is not exact. The time-out function uses its hint and if it computes a hint on one run, an earlier time-out might be created before that time-out expires. This issue simply yields a handle that doesn't get released until that original hint expires. To illustrate, consider that at the time that the time-out function is run, the youngest time-out is 5 minutes in the future. Soon after, a new environment is opened that has a time-out of 1 minute. If this environment becomes idle (and other operations are going on), the time-out function will not release that environment until the original 5 minute hint expires. This is not a problem since the resources will eventually be released.

On a similar note, if a client crashes during an RPC, our reply generates a SIGPIPE, and our server crashes unless we catch it. Using *signal(SIGPIPE, SIG_IGN)* we can ignore it, and the server will go on. This is a call in our *main()* function that we write. Eventually this client's handles would be timed out as described above. We need this only for the unfortunate window of a client crashing during the RPC.

The options below are primarily for control of the program itself. Details relating to databases and environments should be passed from the client to the server, since the server can serve many clients, many environments and many databases. Therefore it makes more sense for the client to set the cache size of its own environment, rather than setting a default cachesize on the server that applies as a blanket to any environment it may be called upon to open. Options are:

- -t to set the default time-out given to an environment.
- -T to set the maximum time-out allowed for the server.
- -L to log the execution of the server process to a specified file.
- -v to run in verbose mode.
- -M to specify the maximum number of outstanding child server processes/threads we can have at any given time. The default is 10. **[We are not yet doing multiple threads/processes.]**

The Client Code

The client code contains all of the supported functions and methods used in this model. There are several methods in the `__db_env` and `__db` structures that currently do not apply, such as the callbacks. Those fields that are not applicable to the client model point to NULL to notify the user of their error. Some method functions remain unchanged, as well such as the error calls.

The client code contains each method function that goes along with the RPC calls described elsewhere. The client library also contains its own version of `db_env_create()`, which does not result in any messages going over to the server (since we do not yet know what server we are talking to). This function sets up the pointers to the correct client functions.

All of the method functions that handle the messaging have a basic flow similar to this:

- Local arg parsing that may be needed
- Marshalling the message header and the arguments we need to send to the server

- Sending the message
- Receiving a reply
- Unmarshalling the reply
- Local results processing that may be needed

Generated Code

Almost all of the code is generated from a source file describing the interface and an *awk* script. This *awk* script generates six (6) files for us. It also modifies one. The files are:

1. Client file - The C source file created containing the client code.
2. Client template file - The C template source file created containing interfaces for handling client-local issues such as resource allocation, but with a consistent interface with the client code generated.
3. Server file - The C source file created containing the server code.
4. Server template file - The C template source file created containing interfaces for handling server-local issues such as resource allocation, calling into the DB library but with a consistent interface with the server code generated.
5. XDR file - The XDR message description file created.
6. Server sed file - A sed script that contains commands to apply to the server procedure file (i.e. the real source file that the server template file becomes) so that minor interface changes can be consistently and easily applied to the real code.
7. Server procedure file - This is the file that is modified by the sed script generated. It originated from the server template file.

The *awk* script reads a source file, *db_server/rpc.src* that describes each operation and what sorts of arguments it takes and what it returns from the server. The syntax of the source file describes the interface to that operation. There are four (4) parts to the syntax:

1. **BEGIN *function version# codetype*** - begins a new functional interface for the given *function*. Each function has a **version number**, currently all of them are at version number one (1). The **code type** indicates to the *awk* script what kind of code to generate. The choices are:
 - **CODE** - Generate all code, and return a status value. If specified, the client code will simply return the status to the user upon completion of the RPC call.
 - **RETCODE** - Generate all code and call a return function in the client template file to deal with client issues or with other returned items. If specified, the client code generated will call a function of the form `__dbcl_<name>_ret()` where `<name>` is replaced with the function name given here. This function is placed in the template file because this indicates that something special must occur on return. The arguments to this function are the same as those for the client function, with the addition of the reply message structure.
 - **NOCLNTCODE** - Generate XDR and server code, but no corresponding client code. (This is used for functions that are not named the same thing on both sides. The only use of this at the moment is `db_env_create` and `db_create`. The environment create call to the server is actually called from the `DBENV->set_server()` method. The `db_create` code exists elsewhere in the library and we modify that code for the client call.)
2. **ARG *RPC-type C-type varname [list-type]*** - each line of this describes an argument to the function. The argument is called *varname*. The **C-type** given is what it should look like in the C code generated, such as **DB ***, **u_int32_t**, **const char ***. The **RPC-type** is an indication about how the RPC request message should be constructed. The RPC-types allowed are described below.
3. **RET *RPC-type C-type varname [list-type]*** - each line of this describes what the server should

return from this procedure call (in addition to a status, which is always returned and should not be specified). The argument is called *varname*. The *C-type* given is what it should look like in the C code generated, such as **DB ***, **u_int32_t**, **const char ***. The *RPC-type* is an indication about how the RPC reply message should be constructed. The RPC-types are described below.

4. **END** - End the description of this function. The result is that when the awk script encounters the **END** tag, it now has all the information it needs to construct the generated code for this function.

The *RPC-type* must be one of the following:

- **IGNORE** - This argument is not passed to the server and should be ignored when constructing the XDR code. **Only allowed for an ARG specification.**
- **STRING** - This argument is a string.
- **INT** - This argument is an integer of some sort.
- **DBT** - This argument is a DBT, resulting in its decomposition into the request message.
- **LIST** - This argument is an opaque list passed to the server (NULL-terminated). If an argument of this type is given, it must have a *list-type* specified that is one of:
 - **STRING**
 - **INT**
 - **ID**.
- **ID** - This argument is an identifier.

So, for example, the source for the DB->join RPC call looks like:

```
BEGIN    dbjoin          1          RETCODE
ARG      ID              DB *        dbp
ARG      LIST            DBC **       curs    ID
ARG      IGNORE          DBC **       dbcpp
ARG      INT             u_int32_t    flags
RET      ID              long         dbcid
END
```

Our first line tells us we are writing the dbjoin function. It requires special code on the client so we indicate that with the RETCODE. This method takes four arguments. For the RPC request we need the database ID from the dbp, we construct a NULL-terminated list of IDs for the cursor list, we ignore the argument to return the cursor handle to the user, and we pass along the flags. On the return, the reply contains a status, by default, and additionally, it contains the ID of the newly created cursor.

Building and Installing

I need to verify with Don Anderson, but I believe we should just build the server program, just like we do for db_stat, db_checkpoint, etc. Basically it can be treated as a utility program from the building and installation perspective.

As mentioned early on, in the section on DB Modifications, we have a single library, but allowing the user to access the client portion by sending a flag to db_env_create(). The Makefile is modified to include the new files.

Testing is performed in two ways. First I have a new example program, that should become part of the example directory. It is basically a merging of ex_access.c and ex_env.c. This example is adequate to test basic functionality, as it does just does database put/get calls and appropriate open and close calls. However, in order to test the full set of functions a more generalized scheme is required. For the

moment, I am going to modify the Tcl interface to accept the server information. Nothing else should need to change in Tcl. Then we can either write our own test modules or use a subset of the existing ones to test functionality on a regular basis.

```

/*-----
*
* File Name: LALStatusMacros.h
*
* Author: Creighton, J. D. E. and Creighton, T. D.
*
* Revision: $Id: LALStatusMacros.h,v 1.2 2000/02/17 19:22:26 jolien Exp $
*-----
*
* NAME
* LALStatusMacros.h
*
* SYNOPSIS
* #include "LALStatusMacros.h"
*
* DESCRIPTION
*
* This header file defines programming macros for handling LAL status
* pointers. The intent is simultaneously to standardize the error
* reporting, and to make the reporting as transparent as possible to
* people coding individual routines.
*
* The following summarized everything the common programmer needs to
* know in order to follow LAL standard error reporting.
*
* 0. The Status structure
*
* All error reporting is handled by a structure type named Status.
* This structure has the following fields:
*
* INT4 statusCode: A code indicating the exit status of a
* function. 0 represents a normal exit. Negative values are
* reserved for certain standard error types. The authors of
* individual functions should assign positive values to the
* various ways in which their code can fail.
*
* const CHAR *statusDescription: An explanatory string
* corresponding to the numerical status code.
*
* const CHAR *Id: A character string identifying the source file
* and version number of the function being reported on.
*
* const CHAR *file: The filename of the actual .c file containing
* the function code.
*
* INT4 line: The line number in the .c file of the instruction
* where any error was reported.
*
* Status *statusPtr: A recursive pointer to another status
* pointer. This structure is used to report an error in a
* subroutine of the current function. Thus if an error occurs
* in a deeply-nested routine, the status structure returned to
* the main program will be the head of a linked list of status
* structures, one for each nested level, with the tail structure
* reporting the actual error that caused the overlying routines
* to fail.
*
* INT4 level: The nested-function level where any error was reported.

```

The standard status codes are as follows:

- 0: Nominal execution: the function returned successfully.
- 1: Recursive error: the function aborted due to failure of a subroutine.
- 2: The status structure passed to the function had a non-null statusPtr field, which blocks the function from calling subroutines (it is symptomatic of something screwy going on in the calling routine).
- 4: Memory allocation error: the function was unable to allocate the statusPtr field to pass down to a subroutine.
- 8: The statusPtr could not be deallocated at the end of all subroutine calls; one of the subroutines must have lost it or set it to null.

1. Every source file should have a unique character string identifying that version of that file. The standard convention, for a file Myfunk.c, is to declare a string at the top of the module using the macro NRCSID (defined in the include file LALRCSID.h):

```
NRCSID (MYFUNKC, "\044Id\044");
```

where \044Id\044 is expanded by RCS to give the full name and version number of the source file.

2. All functions should have return type void. The first argument of any function should be a pointer to a structure of type Status. Thus:

```
void Myfunk(Status *stat, ... )
```

Since the function has no return code, it must report all errors or failure through the status structure. A function that is passed a NULL pointer in place of the status pointer should abort the program, as this is its only way to report the error. However, this is the only circumstance under which a function should normally abort; other errors should be trapped, reported in the status structure, and control returned to the calling routine.

3. The first instruction in any function, after variable declarations, should be the macro INITSTATUS(), which takes two arguments: the function's status pointer, and the module's RCS Id string.

```
INITSTATUS(stat, MYFUNKC);
```

This macro checks that a valid status pointer has been passed to the function, and if so, initializes the other fields to indicate (by default) nominal execution. If stat is null, the macro causes the program to abort, this being its only way to report that something went wrong.

4. Upon completion, the function should issue the macro RETURN(), which takes one argument: the function's status pointer.

```
*
*      RETURN(stat);
*
```

```
*      This takes the place of any return statements, and may also log
*      status reports to some suitable log file (often stderr),
*      depending on implementation and the value of a global debuglevel
*      parameter. Typically RETURN() is used only for successful
*      completion, with other macros ABORT(), ASSERT(), and TRY() being
*      used to report failure. However, it is possible for the
*      programmer to assign the fields of stat by hand, and then issue
*      RETURN(stat).
*
```

- ```
* 5. The standard method to terminate a function unsuccessfully is
* with the ABORT() macro, which takes three arguments: the status
* pointer, the status code, and the status description string.
* Normally the various error codes and descriptions will be
* constants defined in the function's header file:
```

```
* ABORT(stat, MYFUNK_EMYERR, MYFUNK_MSGEMYERR);
*
```

```
* where the error code MYFUNK_EMYERR and the error message
* MYFUNK_MSGEMYERR are defined in the function's header file.
* Like RETURN(), ABORT() correctly handles any status logging
* required by the implementation and the debuglevel. Notably,
* ABORT() does not raise a SIGABRT flag, but instead returns
* control to the calling routine.
*
```

- ```
* 6. Another way to indicate an unsuccessful termination is with the
*     macro ASSERT(), which takes as arguments a test statement, a
*     status pointer, a status code, and a status description. The
*     statement ASSERT(assertion,...); is in all ways equivalent to
*     the statement if(!assertion) ABORT(...);. For instance, in the
*     above example, one might have:
```

```
*      ASSERT(assertion, stat, MYFUNK_EMYERR, MYFUNK_MSGEMYERR)
*
```

```
*      Coding purists may argue that ASSERT() should be used only to
*      trap coding errors rather than runtime errors, which would be
*      trapped using ABORT(). In other words, the assertion should
*      always test true in the final debugged program. At present,
*      however, this coding practice is not enforced by the LAL
*      standard.
*
```

- ```
* 7. If the function is to call other LAL functions as subroutines,
* four more macros must be used to report possible errors arising
* in these routines. The macros are ATTATCHSTATUSPTR(),
* DETATCHSTATUSPTR(), CHECKSTATUSPTR(), and TRY(). The usage of
* these macros is as follows.
```

- ```
*     a. First, before any subroutines are called, the function must
*         call the macro ATTATCHSTATUSPTR() which takes as its argument
*         the status pointer of the current function:
```

```
*      ATTATCHSTATUSPTR(stat);
*
```

```
*      This allocates stat->statusPtr, which is the status pointer
*      that will be handed down into any and all subroutines. If
*      the pointer has already been allocated, ATTATCHSTATUSPTR()
*      will abort, as this is symptomatic of a coding error. Note
*      that ATTATCHSTATUSPTR() need only be called once in a given
```

function, no matter how many subroutine calls that function makes. Normally it should be called immediately after INITSTATUS().

- b. When a subroutine is called, it should be handed the statusPtr field of the calling functions status structure, to report its own errors. The calling function should test the returned status code, and either attempt to deal with any abnormal returns, or abort with status code -1. The macro CHECKSTATUSPTR() helps simplify this procedure. It takes one arguments: the status pointer of the current function (not the subroutine).

```
mysubroutine(stat->statusPtr,...);
CHECKSTATUSPTR(stat);
```

The TRY() macro is a somewhat more streamlined approach but with equivalent results. It takes two arguments. The first is the subroutine call, and the second is the status pointer. Thus:

```
TRY(mysubroutine(stat->statusPtr,...), stat);
```

The only practical difference between these two approaches is that the TRY() macro takes up only one line, and can also report the name of the failed subroutine call when logging errors.

- c. After all subroutines have been called, but before any RETURN() statement, the function must call the DETATCHSTATUSPTR() macro, with the status pointer of the current function (not the subroutines) as its argument:

```
DETATCHSTATUSPTR(stat);
```

This simply deallocates stat->statusPtr and sets it to NULL. It is an error to exit the function with non-NULL statusPtr, unless the exit was due to a subroutine failure. ABORT() and ASSERT() check for this automatically; the only place you need to call DETATCHSTATUSPTR() is immediately before RETURN().

8. The REPORTSTATUS() macro is used to issue a current status report from the current function; the report is printed to stderr or some other implementation-specific stream used for error reporting. REPORTSTATUS() takes the current status pointer as its argument, and iteratively reports down any chain of non-NULL statusPtr structures passed back by subroutines.

9. The top-level main function should start with an empty (all fields set to zero) status structure, as in the following example:

```
int
main ()
{
    static Status status;
    MYFUNK (&status);
    REPORTSTATUS (&status);
    return 0;
}
```


* Non-Conformant Functions:

* These standards apply only to functions that will be publicly available in the LAL libraries. Within a module, a programmer may define and use subroutines that do not conform to the LAL function standards, provided these routines are only visible within that module. Such functions should be declared as static to ensure this. A publicly-visible non-conformant function requires special dispensation.

* Example:

* As an example to illustrate these standards, here is a pair of (rather silly) routines to perform inversion and division of real numbers. In the file division.h one might have the following error codes and error messages defined:

```
* #define DIVISION_ENULL 1
* #define DIVISION_EDIV0 2
* #define DIVISION_MSGEDIV0 "Null pointer"
* #define DIVISION_MSGEDIV0 "Division by zero"
```

* followed by function prototypes. The file division.c might contain the following code:

```
* static const char *DIVISIONC="\044Id\044";
*
* ReturnCode InvertREAL4(Status *stat,
*                       REAL4 *output,
*                       REAL4 input)
* {
*     INITSTATUS(stat,DIVISIONC);
*     ASSERT(output!=NULL,stat,DIVISION_ENULL,DIVISION_MSGENULL);
*     if(input==0.0)
*         ABORT(stat,DIVISION_EDIV0,DIVISION_MSGEDIV0);
*     *output = 1.0/input;
*     RETURN(stat);
* }
*
* ReturnCode DivideREAL4(Status *stat,
*                       REAL4 *output,
*                       REAL4 numerator,
*                       REAL4 denominator)
* {
*     REAL4 invDenom;
*
*     INITSTATUS(stat,DIVISIONC);
*     ATTATCHSTATUSPTR(stat);
*     TRY(InvertREAL4(stat->statusPtr,&invDenom,denominator),stat);
*     *output = numerator*invDenom;
*     DETATCHSTATUSPTR(stat);
*     RETURN(stat);
* }
```

* NOTES

```

* Why are the status handling routines written as macros rather than
* functions?  There are three good reasons.
*
* First, many of the handling routines must be able to force an exit
* from the function calling them.  This cannot be done if the routine
* is in its own function, except by raising signal flags (which is a
* Bad Thing according to LAL standards).
*
* Second, it is useful for these routines to assign a status
* structure's file and line fields using the __FILE__ and __LINE__
* macros.  If the routine is its own function, then these will just
* give the file and line number where the error handling routine is
* defined.  If the routine is a macro, then these will give the file
* and line number where the macro was called, which is much more
* interesting.
*
* Third, by expanding macros at compile time, the runtime performance
* of the resulting code is marginally better.  Most of these macros
* will, under nominal conditions, reduce to a single conditional test
* of an integer value, with no additional overhead from function
* calling and parameter passing.  Thus programmers can be encouraged
* to include extensive error trapping in all their routines, without
* having to worry about compromising performance.
*
*
*-----
*/

```

```

#ifndef _LALSTATUSMACROS_H
#define _LALSTATUSMACROS_H

#ifndef _LALMALLOC_H
#include "LALMalloc.h"
#ifndef _LALMALLOC_H
#define _LALMALLOC_H
#endif
#endif

#ifndef _LALDATATYPES_H
#include "LALDatatypes.h"
#ifndef _LALDATATYPES_H
#define _LALDATATYPES_H
#endif
#endif

#ifndef _LALERROR_H
#include "LALError.h"
#ifndef _LALERROR_H
#define _LALERROR_H
#endif
#endif

#ifndef _LALRCSID_H
#include "LALRCSID.h"
#ifndef _LALRCSID_H
#define _LALRCSID_H
#endif
#endif

```

PROSID (LALSTATUSMACROSH, "\$Id: LALStatusMacros.h,v 1.2 2000/02/17 19:22:26 jolien E

```
extern int debuglevel;
```

```
#define INITSTATUS(statusptr, id) \
do \
{ \
    INT4 level; \
    if(!(statusptr)) \
    { \
        CHAR msg[1024]; \
        sprintf(msg,"Abort: line %d, file %s, %s\n" \
            "          Null status pointer passed to function\n", \
            __LINE__, __FILE__, (id)); \
        LALAbort(msg); \
    } \
    level = (statusptr)->level; \
    memset((statusptr),0,sizeof(Status)); \
    (statusptr)->level = level > 0 ? level : 1 ; \
    (statusptr)->Id     = (id); \
} while (0)
```

```
#define RETURN(statusptr) \
do \
{ \
    (statusptr)->file=__FILE__; \
    (statusptr)->line=__LINE__; \
    if (debuglevel==0 || ((debuglevel==1)&&((statusptr)->statusCode==0))) \
    { \
        return; \
    } \
    else if((statusptr)->statusCode==0) \
    { \
        LALPrintError("Nominal[%d]: line %d, file %s, %s\n", \
            (statusptr)->level, (statusptr)->line, (statusptr)->file, \
            (statusptr)->Id); \
        return; \
    } \
    else \
    { \
        LALPrintError("Error[%d] %d: line %d, file %s, %s\n" \
            "          %s\n", (statusptr)->level, \
            (statusptr)->statusCode, (statusptr)->line, \
            (statusptr)->file, (statusptr)->Id, \
            (statusptr)->statusDescription); \
        return; \
    } \
} while (0)
```

```
#define ABORT(statusptr,code,msg) \
do \
{ \
    (statusptr)->file=__FILE__; \
    (statusptr)->line=__LINE__; \
    (statusptr)->statusCode=(code); \
    (statusptr)->statusDescription=(msg); \
    if((statusptr)->statusPtr) \
    { \
        LALFree((statusptr)->statusPtr); \
        (statusptr)->statusPtr=NULL; \
    } \
} while (0)
```

```

    }
    if(debuglevel==0 || ((debuglevel==1)&&((code)==0)))
    {
        return;
    }
    else if((code)==0)
    {
        LALPrintError("Nominal[%d]: line %d, file %s, %s\n",
            (statusptr)->level, (statusptr)->line, (statusptr)->file,
            (statusptr)->Id);
        return;
    }
    else
    {
        LALPrintError("Error[%d] %d: line %d, file %s, %s\n"
            "          %s\n", (statusptr)->level, (code),
            (statusptr)->line, (statusptr)->file, (statusptr)->Id,
            (mesg));
        return;
    }
} while (0)

#define ASSERT(assertion, statusptr, code, mesg)
do
{
    if(!(assertion))
    {
        (statusptr)->file=__FILE__;
        (statusptr)->line=__LINE__;
        (statusptr)->statusCode=(code);
        (statusptr)->statusDescription=(mesg);
        if((statusptr)->statusPtr)
        {
            LALFree((statusptr)->statusPtr);
            (statusptr)->statusPtr=NULL;
        }
        if(debuglevel==0 || ((debuglevel==1)&&((code)==0)))
        {
            return;
        }
        else if((code)==0)
        {
            LALPrintError("Nominal[%d]: line %d, file %s, %s\n",
                (statusptr)->level, (statusptr)->line,
                (statusptr)->file, (statusptr)->Id);
            return;
        }
        else
        {
            LALPrintError("Error[%d] %d: line %d, file %s, %s\n"
                "          Assertion %s failed: %s\n",
                (statusptr)->level, (code), (statusptr)->line,
                (statusptr)->file, (statusptr)->Id, #assertion,
                (mesg));
            return;
        }
    }
} while (0)

#define ATTATCHSTATUSPTR(statusptr)

```

```

do
{
    ASSERT(!(statusptr)->statusPtr,statusptr,-2,
        "ATTATCHSTATUSPTR: non-null status pointer");
    (statusptr)->statusPtr=(Status *)LALCalloc(1,sizeof(Status));
    ASSERT((statusptr)->statusPtr,statusptr,-4,
        "ATTATCHSTATUSPTR: memory allocation error");
    (statusptr)->statusPtr->level=(statusptr)->level + 1;
} while (0)

#define DETATCHSTATUSPTR(statusptr)
do
{
    ASSERT((statusptr)->statusPtr,statusptr,-8,
        "DETATCHSTATUSPTR: null status pointer");
    LALFree((statusptr)->statusPtr);
    (statusptr)->statusPtr=NULL;
} while (0)

#define TRY(function,statusptr)
do
{
    (function);
    if((statusptr)->statusPtr->statusCode)
    {
        (statusptr)->file=__FILE__;
        (statusptr)->line=__LINE__;
        (statusptr)->statusCode=-1;
        (statusptr)->statusDescription="Recursive error";
        if(debuglevel>0)
        {
            LALPrintError("Error[%d] %d: line %d, file %s, %s\n"
                "          Function call %s failed\n",
                (statusptr)->level,-1,(statusptr)->line,
                (statusptr)->file,(statusptr)->Id,#function);
        }
        return;
    }
} while (0)

#define CHECKSTATUSPTR(statusptr)
do
{
    if((statusptr)->statusPtr->statusCode)
    {
        (statusptr)->file=__FILE__;
        (statusptr)->line=__LINE__;
        (statusptr)->statusCode=-1;
        (statusptr)->statusDescription="Recursive error";
        if(debuglevel>0)
        {
            LALPrintError("Error[%d] %d: line %d, file %s, %s\n"
                "          Function call failed\n",
                (statusptr)->level,-1,(statusptr)->line,
                (statusptr)->file,(statusptr)->Id);
        }
        return;
    }
} while (0)

```

```

#define REPORTSTATUS(statusptr) \
do \
{ \
    Status *ptr; \
    for(ptr=(statusptr);ptr;ptr=(ptr->statusPtr)) \
    { \
        LALPrintError("\nLevel %i: %s\n",ptr->level,ptr->Id); \
        if (ptr->statusCode) \
        { \
            LALPrintError("\tStatus code %i: %s\n",ptr->statusCode, \
                ptr->statusDescription); \
        } \
        else \
        { \
            LALPrintError("\tStatus code 0: Nominal\n"); \
        } \
        LALPrintError("\tfile %s, line %i\n",ptr->file,ptr->line); \
    } \
} while (0)

#endif

```



From:ProQuest

ProQuest Information and Learning Company

[Home](#)[Desktop](#)[Library](#)[Recent Searches](#)[Recent Pages](#)[Notes](#)[Bookmarks](#)[Log](#)

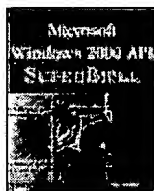
▼ Search

☐ Current Book [Go](#)☐ Code Fragments only[Advanced Search](#)[E-Mail](#) [Add Note](#) [Add Bookmark](#)

▼ Table of Contents



Microsoft Windows
2000 API
SuperBible



- Table of Contents
- Index

Microsoft Windows 2000 API SuperBible

By Richard Simon

Publisher : Sams Publishing

Pub Date : September 13, 2000

ISBN : 0-672-31933-0

Pages : 1608

Slots : 2.0

- ✚ Copyright
- ✚ About the Author
- ✚ Acknowledgments
- ✚ Tell Us What You Think!
- ✚ Introduction
- ± WINDOWS 2000 PROGRAMMING
- ± THE WIN32 API
- ± CREATING WINDOWS
- ± WINDOWS SUPPORT FUNCTIONS
- ± MESSAGE PROCESSING
- ± MENUS
- ± RESOURCES
- ± DIALOG BOXES
- ± USER INPUT FUNCTIONS
- ± SCROLLBARS
- ± MEMORY MANAGEMENT
- ± THE GRAPHICS DEVICE INTERFACE
- ± BITMAPS, ICONS, AND METAFILES
- ± PRINTING AND TEXT OUTPUT
- ± PAINTING AND DRAWING
- ± PALETTES AND COLOR MATCHING
- ± I/O WITH FILES
- ± THE CLIPBOARD

Following in the tradition of the Win 95 and Win32 NT S *Windows 2000 API SuperBible* is a comprehensive reference of Windows APIs needed by the Windows programmer. It is arranged so that related functions are presented in the same chapter. Using related functions that are complete and appropriate to the task at hand, the user can see clearly the most effective usage for each function. The organization of the book will save programmers valuable time by eliminating the need to sift through vendor supplied documentation which is often disorganized. Topics covered include: Windows 2000 Pro Win32 API, the Graphics Device Interface, the Registry, COM (not included in previous versions), Processes, Threads, a Multiple-Document Interface (MDI), and Dynamic Link Library

URL <http://proquest.safaribooksonline.com/0672319330>

Book: Microsoft Windows 2000 API SuperBible**Section:** Chapter 25. PROCESSES, THREADS, AND FIBERS**Process, Thread, and Fiber Function Descriptions**

Table 25.3 lists the process, thread, and fiber functions of the Win32 API. Complete function descriptions follow the table.

Table 25.3. Process, Thread, and Fiber Function Summary

Function	Purpose
AttachThreadInput	Redirects input of one thread to another thread.
CancelWaitableTimer	Cancels a waitable timer by setting the timer to the inactive state.
ConvertThreadToFiber	Converts the current thread into a fiber.
CreateEvent	Creates an event object.
CreateFiber	Creates a fiber object.
CreateMutex	Creates a mutex object.
CreateProcess	Creates a process and thread object and starts a process.
CreateProcessAsUser	Creates a process and thread object in the security context of a user.
CreateRemoteThread	Creates a thread that runs in the address space of another process.
CreateSemaphore	Creates a semaphore object.
CreateThread	Creates a thread object and starts a thread.
CreateWaitableTimer	Creates a waitable timer object.
DeleteCriticalSection	Destroys an inactive critical section object.
DeleteFiber	Deletes a fiber object.
DuplicateHandle	Copies the handle of any object so that it can be used by another process.
EnterCriticalSection	Tests a critical section gate, and if unowned, the critical section is entered.
ExitProcess	Ends a process and all the process's threads and returns an exit code.
ExitThread	Ends a single thread of a process.
GetCurrentFiber	Returns the current fiber for the thread.
GetCurrentProcess	Returns the handle of the current process.
GetCurrentProcessId	Returns the ID of the current process.
GetCurrentThread	Returns the handle of the current thread.
GetCurrentThreadId	Returns the ID of the current thread.
GetExitCodeProcess	Returns the exit code of the specified process. Can also be used to determine if the process has been completed.
GetExitCodeThread	Returns the exit code of the specified thread. Can also be used to determine if the given thread has completed.

GetFiberData	Returns the data associated with the fiber object.
GetPriorityClass	Determines the priority class of a process.
GetProcessAffinityMask	Returns the affinity mask for a process.
GetProcessHeap	Returns a handle to the calling process's heap.
GetProcessHeaps	Returns an array of handles that contains all the heaps owned by the calling process.
GetProcessPriorityBoost	Returns the priority boost for a process.
GetProcessShutdownParameters	Returns the shutdown parameters for a process.
GetProcessTimes	Returns the times the process has been executing.
GetProcessVersion	Returns the Windows version the process expects.
GetProcessWorkingSetSize	Returns the working set size for a process.
GetQueueStatus	Returns the input state of a thread's queue.
GetThreadContext	Used by a debugger to store the machine context of a thread.
GetThreadPriority	Returns the priority class of the given thread.
GetThreadPriorityBoost	Returns the priority boost for a thread.
GetThreadSelectorEntry	Used by debuggers to return the local descriptor table (LDT) entry for a given selector of the specified thread.
GetThreadTimes	Returns the times a thread has been executing.
InitializeCriticalSection	Constructs a critical section object to protect a section of a multithreaded process from execution by more than one thread at a time.
InterlockedCompareExchange	Compares the values of a synchronized long.
InterlockedDecrement	Decrements a synchronized long.
InterlockedExchange	Exchanges a value with the value of a synchronized long.
InterlockedExchangeAdd	Adds a value to a synchronized long.
InterlockedIncrement	Increments a synchronized long.
LeaveCriticalSection	Marks the end of code protected by a critical section object.
MsgWaitForMultipleObjects	Returns when any or all of the given objects become signaled, a given type of input appears in the specified thread's input queue, or a time-out occurs.
MsgWaitForMultipleObjectsEx	Same as MsgWaitForMultipleObjects, except that an I/O completion or APC causes the function to also return.
OpenEvent	Returns the handle of a named event object.
OpenMutex	Returns the handle of a named mutex object.
OpenProcess	Returns the handle of a specified process.
OpenSemaphore	Returns the handle of a named semaphore object.
OpenWaitableTimer	Returns the handle of a waitable timer object.
PulseEvent	Changes an event from nonsignaled to signaled and then to nonsignaled again. Used to free waiting threads and then set the event block again.
QueueUserAPC	Queues a user-mode asynchronous procedure call (APC) object to the APC queue

Returns **BOOL:** If successful, TRUE is returned; otherwise, FALSE is returned.

Include File winbase.h

See Also SetProcessShutdownParameters

GetProcessTimes

■ Windows 98 ■ Windows 2000

Description GetProcessTimes retrieves timing information about a specified process. All times are expressed using FILETIME data structures. For a definition of this structure, see the CompareFileTime function in Chapter 17, "I/O with Files." Process creation and exit times are points in time expressed as the amount of time that has elapsed since midnight on January 1, 1601, GMT. Use the Win32 API functions to convert such values to more useful forms. Process kernel mode and user mode times are amounts of time elapsed.

Syntax **BOOL** GetProcessTimes(**HANDLE** *hProcess*, **LPFILETIME** *lpCreationTime*, **LPFILETIME** *lpExitTime*, **LPFILETIME** *lpKernelTime*, **LPFILETIME** *lpUserTime*)

Parameters

hProcess **HANDLE:** An open handle that specifies the process whose timing information is retrieved. This handle must be created with PROCESS_QUERY_INFORMATION access.

lpCreationTime **LPFILETIME:** A pointer to a FILETIME structure that receives the creation time of the process.

lpExitTime **LPFILETIME:** A pointer to a FILETIME structure that receives the exit time of the process. If the process has not exited, the content of this structure is undefined.

lpKernelTime **LPFILETIME:** A pointer to a FILETIME structure that receives the amount of time that the process has executed in kernel mode. This is the sum of the time that each thread owned by the process has spent in kernel mode.

lpUserTime **LPFILETIME:** A pointer to a FILETIME structure that receives the amount of time that the process has executed in user mode. This is the sum of the time that each thread owned by the process has spent in user mode.

Returns **BOOL:** If successful, TRUE is returned; otherwise, FALSE is returned. Use the GetLastError function to retrieve extended error information.

Include File winbase.h

See Also FileTimeToDosDateTime, FileTimeToLocalFileTime, FileTimeToSystemTime

GetProcessVersion

■ Windows 98 ■ Windows 2000

Description GetProcessVersion retrieves the major and minor version numbers of the Windows version on which a specified process expects to run. The version number returned by this function is the version number stamped in the image header of the EXE file the process is running.

Syntax **DWORD** GetProcessVersion(**DWORD** *ProcessId*)

Parameters

ProcessId **DWORD:** The process identifier of the process for which to retrieve the version information. If set to 0, the calling process is used.

Returns **DWORD:** If successful, the version of Windows on which the process expects to run is returned. The high word of the return value contains the major version number. The low word of the return value contains the minor version number. If an error occurs, 0 is returned. Use the GetLastError to retrieve extended error information.

Include File winbase.h

See Also [GetProcessWorkingSetSize](#)

GetProcessWorkingSetSize

■ Windows 98 ■ Windows 2000

Description `GetProcessWorkingSetSize` retrieves the minimum and maximum *working set* sizes, in bytes, of a specified process. The working set of a process is the set of memory pages currently visible to the process in physical RAM memory. These sizes affect the virtual memory paging behavior of a process.

Syntax `BOOL GetProcessWorkingSetSize(HANDLE hProcess, LPDWORD lpMinWorkingSetSize, LPDWORD lpMaxWorkingSetSize)`

Parameters

hProcess **HANDLE:** An open handle to the process for which to retrieve the working set sizes. The handle must have the `PROCESS_QUERY_INFORMATION` access rights.

lpMinWorkingSetSize **LPDWORD:** A pointer to a variable that receives the minimum working set size, in bytes, of the specified process. The virtual memory manager attempts to keep at least this much memory resident in the process whenever the process is active.

lpMaxWorkingSetSize **LPDWORD:** A pointer to a variable that receives the maximum working set size, in bytes, of the specified process. The virtual memory manager attempts to keep no more than this much memory resident in the process whenever the process is active when memory is in short supply.

Returns **BOOL:** If successful, `TRUE` is returned; otherwise, `FALSE` is returned. Use the `GetLastError` function to retrieve extended error information.

Include File `winbase.h`

See Also [SetProcessWorkingSetSize](#)

GetQueueStatus

■ Windows 98 ■ Windows 2000

Description `GetQueueStatus` returns the status of the calling thread's message queue. The status indicates the type of messages found in the queue. The presence of a `QS_` flag in the return value does not guarantee that a subsequent call to the `PeekMessage` or `GetMessage` function will return a message. These functions perform some internal filtering that can cause the message to be processed internally.

Syntax `DWORD GetQueueStatus(UINT uFlags)`

Parameters

uFlags **UINT:** Indicates the type of messages for which to check. This can be one or more of the values listed in Table 25.10.

Table 25.10. Queue Status Types

Value	Meaning
<code>QS_ALLEVENTS</code>	An input message (<code>WM_TIMER</code> , <code>WM_PAINT</code> , <code>WM_HOTKEY</code>) or posted message is in the queue.
<code>QS_ALLINPUT</code>	Any message is in the queue.
<code>QS_ALLPOSTMESSAGE</code>	A posted message is in the queue.
<code>QS_HOTKEY</code>	A <code>WM_HOTKEY</code> message is in the queue.
<code>QS_INPUT</code>	An input message is in the queue (<code>QS_KEY</code> or <code>QS_MOUSE</code>).

File 275:Gale Group Computer DB(TM) 1983-2004/May 03
 (c) 2004 The Gale Group
 File 621:Gale Group New Prod.Annou.(R) 1985-2004/Apr 30
 (c) 2004 The Gale Group
 File 636:Gale Group Newsletter DB(TM) 1987-2004/May 03
 (c) 2004 The Gale Group
 File 16:Gale Group PROMT(R) 1990-2004/May 03
 (c) 2004 The Gale Group
 File 160:Gale Group PROMT(R) 1972-1989
 (c) 1999 The Gale Group
 File 148:Gale Group Trade & Industry DB 1976-2004/May 03
 (c)2004 The Gale Group
 File 624:McGraw-Hill Publications 1985-2004/May 03
 (c) 2004 McGraw-Hill Co. Inc
 File 15:ABI/Inform(R) 1971-2004/May 01
 (c) 2004 ProQuest Info&Learning
 File 647:CMP Computer Fulltext 1988-2004/Apr W4
 (c) 2004 CMP Media, LLC
 File 674:Computer News Fulltext 1989-2004/Apr W4
 (c) 2004 IDG Communications
 File 696:DIALOG Telecom. Newsletters 1995-2004/Apr 30
 (c) 2004 The Dialog Corp.
 File 369:New Scientist 1994-2004/Apr W4
 (c) 2004 Reed Business Information Ltd.

Set	Items	Description
S1	13809	FUNCTION? ?(7N)VERSION
S2	158623	SOURCE()CODE OR PROGRAMMING()LANGUAGE? ? OR OOPL
S3	479	S1(100N)S2
S4	3427	VERSION() (NUMBER OR NO OR ID)
S5	89	S4(10N)FUNCTION? ?
S6	4	S5(100N)S2
S7	62	RD S5 (unique items)

7/3,K/1 (Item 1 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

02458986 SUPPLIER NUMBER: 67870891 (USE FORMAT 7 OR 9 FOR FULL TEXT)

IE Tracks Down the Bugs.

Methvin, Dave

WinMag.com, NA

Dec 11, 2000

LANGUAGE: English RECORD TYPE: Fulltext

WORD COUNT: 831 LINE COUNT: 00062

TEXT:

...to Microsoft to help them debug the problem. MS Support article Q276550 describes the basic **function** of the software. The new **version number** they give for IE only applies if you have IE 5.5, the numbers will ...

7/3,K/2 (Item 2 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

02458656 SUPPLIER NUMBER: 67974005 (USE FORMAT 7 OR 9 FOR FULL TEXT)

IE Tracks Down the Bugs. (Product Support) (Tutorial)

Methvin, Dave

WinMag.com, NA

Dec 13, 2000

DOCUMENT TYPE: Tutorial LANGUAGE: English RECORD TYPE: Fulltext;

Abstract

WORD COUNT: 831 LINE COUNT: 00062

TEXT:

...to Microsoft to help them debug the problem. MS Support article Q276550 describes the basic **function** of the software. The new **version number** they give for IE only applies if you have IE 5.5, the numbers will ...

7/3,K/3 (Item 3 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

02384243 SUPPLIER NUMBER: 60369844 (USE FORMAT 7 OR 9 FOR FULL TEXT)

Gravity Storm Tackles Service Packs. (Product Announcement)

ENT, 5, 1, 12

Jan 12, 2000

DOCUMENT TYPE: Product Announcement ISSN: 1085-2395 LANGUAGE:

English RECORD TYPE: Fulltext

WORD COUNT: 206 LINE COUNT: 00020

... workstation. Checked off machines appear in a second pane showing the machine's Windows NT **version number** ; **function** ; Service Pack number; Service Pack download date; and encryption strength. Clicking on a machine in...

7/3,K/4 (Item 4 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

02109480 SUPPLIER NUMBER: 19769757 (USE FORMAT 7 OR 9 FOR FULL TEXT)

Build dynamic web pages: beyond CGI. (Microsoft NSAPI and Netscape ISAPI are more flexible than CGI) (Internet/Web/Online Service Information)

McClanahan, David

Databased Web Advisor, v15, n9, p46(6)

Sep, 1997

ISSN: 1090-6436 LANGUAGE: English RECORD TYPE: Fulltext; Abstract

WORD COUNT: 3796 LINE COUNT: 00346

... HttpExtensionProc(). For ISAPI filters the corresponding functions are GetFilterVersion() and HttpFilterProc(). IIS calls the GetxxxVersion() **function** at initialization to get the .DLL's **version number** and description string. IIS calls the HttpxxxProc() **function** whenever a user (a web page) invokes this ISAPI application. The HttpxxxProc() function is the...

7/3,K/5 (Item 5 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02062153 SUPPLIER NUMBER: 19387792 (USE FORMAT 7 OR 9 FOR FULL TEXT)
BindView Scores With NOSadmin. (BindView Development NOSadmin for Windows NT 5.0 network management software) (PC Week Netweek) (Software Review) (Evaluation)
Sturdevant, Cameron
PC Week, v14, n18, p115(2)
May 5, 1997
DOCUMENT TYPE: Evaluation ISSN: 0740-1604 LANGUAGE: English
RECORD TYPE: Fulltext; Abstract
WORD COUNT: 1260 LINE COUNT: 00105

... older siblings NOSadmin for NetWare 3.x and 4.x; hence the reason for its **version number**. The tool fills a specific and somewhat unique network management **function** by providing live access to NT server and domain configuration information through centralized reports, allowing...

7/3,K/6 (Item 6 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02036287 SUPPLIER NUMBER: 19054521 (USE FORMAT 7 OR 9 FOR FULL TEXT)
E-mail: old meets new. (proprietary and Internet e-mail offerings) (Technology Information)
Hurwicz, Mike
LAN Magazine, v12, n2, p87(5)
Feb, 1997
ISSN: 1069-5621 LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 4680 LINE COUNT: 00391

... 5, released at the end of 1996, rounds out those offerings, adding calendaring and scheduling **functions**.

Microsoft Exchange Server 4.0--despite its **version number**--is a first-generation product that was introduced only last year. Though it is relatively...

7/3,K/7 (Item 7 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02031802 SUPPLIER NUMBER: 19096785
Under the hood. (bypassing empty drives in multiple CD-ROM drives) (Technology Tutorial)
D'Agostino, Peter
Windows Tech Journal, v6, n2, p34(4)
Feb, 1997
ISSN: 1061-3501 LANGUAGE: English RECORD TYPE: Abstract

...ABSTRACT: functions of MSCDEX used to identify empty CD drives include the CD-ROM Drive Check **function**, the MSCDEX **version number** and the Send Device Request.

7/3,K/8 (Item 8 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01966109 SUPPLIER NUMBER: 18537281
Q & A C++. (Question and Answer)
DiLascia, Paul
Microsoft Systems Journal, v11, n9, p103(4)
Sep, 1996
ISSN: 0889-9932 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 2462 LINE COUNT: 00190

... MFC code is compiled into your EXE file. In the latter case, your code calls **functions** in a DLL MFCxxD.DLL, where xx is the **version number** and D indicates the debug version of the DLL. For example, my \WINDOWS \SYSTEM directory...

7/3,K/9 (Item 9 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01933827 SUPPLIER NUMBER: 18252925 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Roll your own persistence implementations to go beyond the MFC frontier.
(developing a persistence implementation that is not integrated with MFC, but that can coexist with MFC) (Technology Tutorial) (Tutorial)
Holub, Allen
Microsoft Systems Journal, v11, n6, p31(16)
June, 1996
DOCUMENT TYPE: Tutorial ISSN: 0889-9932 LANGUAGE: English
RECORD TYPE: Fulltext; Abstract
WORD COUNT: 7336 LINE COUNT: 00593

... first argument. The second argument tells you the direction (loading from or flushing to disk). **Version - number** support is provided here by means of a virtual **function** (the MFC implementation used the schema argument to the IMPLEMENT...

...SERIAL macro). Override the version **function** with one that returns the correct **version number** for the class definition. This way an error will be returned if you try to...out a 126-byte block that contains the class name followed by a two-byte **version number**. It then calls your **Serialize function** to get the data out on the disk. Remember that all persistent objects derive from...

7/3,K/10 (Item 10 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01853941 SUPPLIER NUMBER: 17507761 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Network management. (LAN Buyers Guide Issue) (Buyers Guide)
LAN Magazine, v10, n10, p201(51)
Oct 15, 1995
DOCUMENT TYPE: Buyers Guide ISSN: 1069-5621 LANGUAGE: English
RECORD TYPE: Fulltext; Abstract
WORD COUNT: 48741 LINE COUNT: 04035

... card throughput, protocols bound to each NIC and frame types used, and NOSs the current **version number**. Control **functions** include remote command execution and threshold settings. It also works with NMS, SunNet Manager, HP...

7/3,K/11 (Item 11 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

1453035 SUPPLIER NUMBER: 17424976 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Building sticky windows. (providing persistent window states under Windows
95) (PC Tech: Power Programming) (Column) (Tutorial)
Prosise, Jeff
PC Magazine, v14, n19, p355(6)
Nov 7, 1995
DOCUMENT TYPE: Column Tutorial ISSN: 0888-8507 LANGUAGE: English
RECORD TYPE: Fulltext; Abstract
WORD COUNT: 3722 LINE COUNT: 00376

... defaults to "Version 1.0." If you'd like, you could add a
SetVersionNumber member function to CSmartFrameWnd allowing the version
number to be set programmatically without modifying CSmartFrameWnd's
source code.

The approach that CSmartFrameWnd takes...

7/3,K/12 (Item 12 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01837991 SUPPLIER NUMBER: 17454773 (USE FORMAT 7 OR 9 FOR FULL TEXT)
HTTP Server for Windows NT. (version 0.96; European Microsoft Windows NT
Academic Centre) (one of five evaluations of World Wide Web server
software in "Windows NT Based Servers What's The Rush") (Software
Review) (Evaluation)
Reichard, Kevin
PC Magazine, v14, n17, p239(1)
Oct 10, 1995
DOCUMENT TYPE: Evaluation ISSN: 0888-8507 LANGUAGE: English
RECORD TYPE: Fulltext; Abstract
WORD COUNT: 838 LINE COUNT: 00071

... for freeware, the documentation is remarkably complete.) Map files
are fully supported using the ISMAP function. And as for the reliability
issue, even though the version number is 0.96, the EMWAC HTTP server
has a reputation for being a reliable, low...

7/3,K/13 (Item 13 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01832965 SUPPLIER NUMBER: 17103179 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Conflict Catcher 3. (Casady & Greene extension-management utility)
(Software Review) (Evaluation)
Schorr, Joseph
Macworld, v12, n9, p69(1)
Sep, 1995
DOCUMENT TYPE: Evaluation ISSN: 0741-8647 LANGUAGE: English
RECORD TYPE: Fulltext; Abstract
WORD COUNT: 617 LINE COUNT: 00054

... on any listed file displays even more detailed information about
it, such as its icon, version number, type/creator code, and a
description of its function.

The program lets you color-code start-up files in list views--you can
have...

7/3,K/14 (Item 14 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01820659 SUPPLIER NUMBER: 17385086 (USE FORMAT 7 OR 9 FOR FULL TEXT)
A VxD for system profiling. (Tutorial)
Plooy, Ton
Windows-DOS Developer's Journal, v6, n7, p39(8)

July, 1995

DOCUMENT TYPE: Tutorial ISSN: 1059-2407 LANGUAGE: English

RECORD TYPE: Fulltext; Abstract

WORD COUNT: 2567 LINE COUNT: 00193

... Device macro. irq0.asm exports two **functions** , one to get the **version number** and the other to install a DLL callback **function** for the interrupt handler to call.

The third part of the code is the interrupt...

7/3,K/15 (Item 15 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

01790329 SUPPLIER NUMBER: 16440469 (USE FORMAT 7 OR 9 FOR FULL TEXT)

The ultimate Windows version detector. (Windows NT programming techniques)

(Technical) (Tutorial)

Tomlinson, Paula

Windows-DOS Developer's Journal, v6, n2, p7(11)

Feb, 1995

DOCUMENT TYPE: Technical Tutorial ISSN: 1059-2407 LANGUAGE:

ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT

WORD COUNT: 2826 LINE COUNT: 00222

TEXT:

...the previous version, with backwards compatibility fully intact. You could simply call the Windows API **function** GetVersion() to obtain the current Windows **version number** , which would be all you'd need to know about which operating system you were...

... shell of some kind. For completeness, dostest.exe displays the information returned from INT 21h, **function** 30h.

Next, dostest.exe displays the **version number** retrieved by INT 21h, **function** 3306h. Recall that INT 33h, subfunctions 00h and 01h are used to get and set...

7/3,K/16 (Item 16 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

01724274 SUPPLIER NUMBER: 16114649 (USE FORMAT 7 OR 9 FOR FULL TEXT)

System Architect 3.0. (Popkin Software & Systems' CASE tool) (Software Review) (Evaluation)

Linthicum, David

DBMS, v8, n1, p62(4)

Jan, 1995

DOCUMENT TYPE: Evaluation ISSN: 1041-5173 LANGUAGE: ENGLISH

RECORD TYPE: FULLTEXT; ABSTRACT

WORD COUNT: 1951 LINE COUNT: 00158

...ABSTRACT: development is supported either through merging on single-user versions or by using the network **version** . No code-generation **function** is provided, but a convenient reverse-engineering facility is an excellent tool for reading existing...

7/3,K/17 (Item 17 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

01712501 SUPPLIER NUMBER: 16245671 (USE FORMAT 7 OR 9 FOR FULL TEXT)

The origins of DOS: DOS creator gives his view of relationship between CP/M, MS-DOS. (Tim Paterson) (Letter to the Editor)

Paterson, Tim

Microprocessor Report, v8, n13, p23(3)

Oct 3, 1994

DOCUMENT TYPE: Letter to the Editor ISSN: 0899-9341 LANGUAGE:

ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 2461 LINE COUNT: 00182

... every last CP/M function call. For example, MS-DOS did not implement CP/M **function 12 (0CH)** to get the system **version number**. Somewhat unaccountably, MS-DOS instead used (and still uses) **function 0CH** to read the keyboard." This one change would likely have caused most CP/M ...

7/3,K/18 (Item 18 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01710804 SUPPLIER NUMBER: 16181702 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Windows questions & answers. (Column)
Bonneau, Paul
Windows-DOS Developer's Journal, v5, n8, p6(10)
August, 1994
DOCUMENT TYPE: Column ISSN: 1059-2407 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 2754 LINE COUNT: 00248

... and 3. Function 0 is the easy one, since all it does is return the **version number** of the VxD in the AX register. **Function 3** is used to create a VM.

Function 3 accepts a far pointer to a...

7/3,K/19 (Item 19 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01697677 SUPPLIER NUMBER: 16197462 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Investigating the hybrid windowing and messaging architecture of Chicago. (Microsoft's Chicago operating system) (Technical)
Pietrek, Matt
Microsoft Systems Journal, v9, n9, p15(14)
Sept, 1994
DOCUMENT TYPE: Technical ISSN: 0889-9932 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 12182 LINE COUNT: 00926

... rationale behind this behavior is that many Windows 3.x programs didn't check the **version number** correctly, and broke when the Chicago **GetVersion function** returned 4.0. After studying many programs, the Microsoft coders concluded that returning version 3 up for this kludge, Microsoft included a **GetVersionEx function** in Chicago that is promised to always return the correct **version number**.

In Win32-based programs, the meaning of the high WORD returned from **GetVersion** has been...

7/3,K/20 (Item 20 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01693979 SUPPLIER NUMBER: 16176386 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Assign your own version control. (determining version control numbers for legacy database files) (Integration Database Tools) (Column) (Tutorial)
Watterson, Karen
Windows Sources, v2, n9, p187(2)
Sept, 1994
DOCUMENT TYPE: Tutorial ISSN: 1065-9641 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1362 LINE COUNT: 00100

... 4.0 databases often require different ODBC drivers.

While Paradox for Windows comes with a **function** called VERSION() that returns a string containing the **version number** of Paradox currently being used, the program does not provide an equivalent way to find...

7/3,K/21 (Item 21 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01685398 SUPPLIER NUMBER: 15512343 (USE FORMAT 7 OR 9 FOR FULL TEXT)
More power for WINCMD. (PC Tech) (Utilities) (includes related articles on new features in WINCMD 1.3 and WCLIB 1.1, guide to the utilities and how to obtain them) (Tutorial)

Boling, Douglas
PC Magazine, v13, n12, p282(7)
June 28, 1994

DOCUMENT TYPE: Tutorial ISSN: 0888-8507 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 6016 LINE COUNT: 00466

... present by using the LIBVER command. LIBVER's single parameter is the name of a **function** library, such as WCLIB or WCL2. The **function** returns the **version number** of the library or 0 if the library is not loaded.

An example of a...

7/3,K/22 (Item 22 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01680875 SUPPLIER NUMBER: 15343724 (USE FORMAT 7 OR 9 FOR FULL TEXT)
A VxD to monitor DOS output. (includes related article on acquiring a VxD ID from Microsoft Corp) (Tutorial)

Bonneau, Paul
Windows-DOS Developer's Journal, v5, n5, p23(18)
May, 1994

DOCUMENT TYPE: Tutorial ISSN: 1059-2407 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 6081 LINE COUNT: 00471

... point is stored in the global variable lpfnVxD. LibMain() then calls through lpfnVxD, placing the **function** number apiGetVersion in AX, to obtain the **version number** of the DLL. If it is less than the version number the DLL was built...

7/3,K/23 (Item 23 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01680874 SUPPLIER NUMBER: 15343658 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Virtualizing a DOS device driver with a VxD. (includes related article on Program Segment Prefixes) (Tutorial)

Tomlinson, Paula
Windows-DOS Developer's Journal, v5, n5, p6(14)
May, 1994

DOCUMENT TYPE: Tutorial ISSN: 1059-2407 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 6055 LINE COUNT: 00478

... with what request sizes. For demonstration purposes, TestVxD calls the Stealth VxD's get version **function** and displays both the **version number** (returned in AX) and the last read length (returned in BX as a debugging variable...

7/3,K/24 (Item 24 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01680279 SUPPLIER NUMBER: 15355996 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Switch Windows setups with Switcher. (PC Magazine's utility) (includes
related articles on top 10 utilities and sources of PC Magazine's
utilities) (PC Tech: Utilities) (Column)
Boling, Douglas
PC Magazine, v13, n10, p311(6)
May 31, 1994
DOCUMENT TYPE: Column ISSN: 0888-8507 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 3913 LINE COUNT: 00305

... for Windows programs. Some items cannot be set--for example, the
mouse type and driver **version number**. If you try to set these items,
the **function** returns an error code. The final action, GetProfileFilename,
is handy for locating the mouse-profile...

7/3,K/25 (Item 25 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01618490 SUPPLIER NUMBER: 14397480 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Writing DOS utilities with DEBUG, part 2. (Tutor) (Column) (Tutorial)
Prosise, Jeff
PC Magazine, v12, n17, p347(4)
Oct 12, 1993
DOCUMENT TYPE: Tutorial ISSN: 0888-8507 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2308 LINE COUNT: 00169

...ABSTRACT: DOS function 2Ah returns the current date. DOS function 2Ch
returns the current time. DOS **function 30h** returns the DOS **version
number**. Although these are only three of the many programming services
available through DOS, they are...
... which returns the current date; DOS function 2Ch, which returns the
current time; and DOS **function 30h**, which returns the DOS **version
number**. These are but three of the dozens of programming services that DOS
provides, but they...

...The instruction CMP AL,3 compares the value in register AL (which holds
the major **version number** after a call to DOS **function 30h**) to 3. Then
the instruction JB 0180 jumps to 0180h if the value in...

...MOV AH,4C
INT 21

N GETVER.COM
RCX
8
W
Q

GETVER calls DOS **function 30h** to get the major **version number** in
AL. Then it invokes DOS **function 4Ch** to terminate itself. Recall from the
last issue that when function 4Ch is invoked...back to GETVER2.COM. After
the major and minor version numbers are obtained by calling **function 30h**,
the instruction MOV DH,AH transfers the minor **version number** to the DH
register, which serves as a temporary storage location. Next, the
instructions
MOV...

7/3,K/26 (Item 26 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01534125 SUPPLIER NUMBER: 13422064 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Add-ons extend Visual Basic's capabilities. (Software Review) (PC Week
Supplement: Development Tools) (Evaluation)
Sullivan, Eamonn
PC Week, v10, n6, pS9(2)
Feb 15, 1993
DOCUMENT TYPE: Evaluation ISSN: 0740-1604 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1531 LINE COUNT: 00121

... C programmers who are developing custom controls. For example, we
could use one of the **functions** to get the **version number** of Visual
Basic or to detect when a Visual Basic control changes the Windows palette
...

7/3,K/27 (Item 27 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01550637 SUPPLIER NUMBER: 13037415 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Documentation and on-line help make good partners. (Software Buyer's Guide:
Project Managers) (Installation, Help and Support) (includes related
article on buying tips)
Gilliland, Steve
PC Sources, v3, n12, p453(2)
Dec, 1992
ISSN: 1052-6579 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1114 LINE COUNT: 00087

... by a pull-down menu, it isn't context-sensitive, and offers nothing
but a **function** -key list; the **version number** of the project; and a
tutorial choice that drops you out of the project on...

7/3,K/28 (Item 28 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01549597 SUPPLIER NUMBER: 13039963 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Identify the running DOS application from Windows. (Tutorial)
Bonneau, Paul
Windows-DOS Developer's Journal, v3, n12, p26(10)
Dec, 1992
DOCUMENT TYPE: Tutorial ISSN: 1059-2407 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 3003 LINE COUNT: 00223

... code for the VxD, DosName. DosName provides three protected-mode
API services: return the VxD **version number**, register a callback
function, and get the name of the current task in a DOS session identified
by VM...

7/3,K/29 (Item 29 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01541038 SUPPLIER NUMBER: 12741205 (USE FORMAT 7 OR 9 FOR FULL TEXT)
R:BASE 3.1D: 4.0 for your 286. (data base management system, Intel 80286
microprocessor) (Brief Article)
Downall, Bill
Data Based Advisor, v10, n10, p77(2)
Oct, 1992
DOCUMENT TYPE: Brief Article ISSN: 0740-5200 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 559 LINE COUNT: 00040

... to check for R:BASE version and branch accordingly. Your code should check for the **version number** with the (CVAL('Version')) **function** . Code for 3.1D users should still ZIP RETURN, but 4.0 users should simply...

7/3,K/30 (Item 30 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01535961 SUPPLIER NUMBER: 12658521 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Tracking file versions: here's a Turbo Pascal utility that will automatically add version numbers to your filenames. (Solutions: Languages) (Column)
Terdeman, Sharon
PC Magazine, v11, n17, p414(2)
Oct 13, 1992
DOCUMENT TYPE: Column ISSN: 0888-8507 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 440 LINE COUNT: 00033

... we must forfeit extensions for such files because there's nowhere else to keep the **version number** .
As originally written, RENVER renamed files by using the Exec **function** to invoke a secondary copy of COMMAND.COM. Since Turbo Pascal has its own Rename...

7/3,K/31 (Item 31 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01519885 SUPPLIER NUMBER: 12202682 (USE FORMAT 7 OR 9 FOR FULL TEXT)
A smooth segue between Clipper and NetWare. (HRF Associates Inc.'s SEGUE application programming interface) (Software Review) (Evaluation)
Mueller, John
Data Based Advisor, v10, n4, p18(2)
April, 1992
DOCUMENT TYPE: Evaluation ISSN: 0740-5200 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1318 LINE COUNT: 00125

TEXT:

...at the back of the book. In addition, each function tells you < what Novell API **function** is being implemented, as well as the API **version number** in which the **function** first appeared. However, one thing the descriptions were lacking was an example of how to...

7/3,K/32 (Item 32 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01501035 SUPPLIER NUMBER: 11987265 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Readers' forum. (subjects discussed include terminate-and-stay-resident programs, Hungarian notation, programming problems and the journal name) (Letter to the Editor)
Edmead, Mark T.; McNamee, John; Rubin, Joel M.; Tilton, Homer B.; Wilson, Chris
Windows-DOS Developer's Journal, v3, n2, p76(4)
Feb, 1992
DOCUMENT TYPE: Letter to the Editor ISSN: 1059-2407 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 1878 LINE COUNT: 00140

... for the DOS version number to be returned to this program if the Get DOS **Version Number function** is called by the program. Evidently,

this is part of the SETVER system.
I was...

7/3,K/33 (Item 33 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01496432 SUPPLIER NUMBER: 11719816 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Questions & answers: MS-DOS. (Column)
Prosise, Jeff
Microsoft Systems Journal, v7, n1, p119(4)
Jan-Feb, 1992
DOCUMENT TYPE: Column ISSN: 0889-9932 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 2657 LINE COUNT: 00193

... DOS version that's running (for example, the 5 in 5.0) and the
minor **version number** in BH. On the surface, this might seem
unnecessary, since **function 30H** also returns version information.
However, function 3306H is not affected by the SETVER command...

7/3,K/34 (Item 34 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01496430 SUPPLIER NUMBER: 11719812 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Creating graphics import filter libraries for Windows applications.
(Tutorial)
Welch, Kevin P.
Microsoft Systems Journal, v7, n1, p101(12)
Jan-Feb, 1992
DOCUMENT TYPE: Tutorial ISSN: 0889-9932 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 3542 LINE COUNT: 00283

... filter has not been previously loaded, the application loads the
library and calls the GetFilterInfo **function**. The application passes
GetFilterInfo its **version number** and a pointer to a copy of the
optional configuration data from the filter's WIN.INI entry. The
GetFilterInfo **function** then checks the **version number** and uses the
optional configuration data to define conversion parameters. If the filter
displays a...

7/3,K/35 (Item 35 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01496427 SUPPLIER NUMBER: 11719806 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Strategies and techniques for writing state-of-the-art TSRs that exploit
MS-DOS 5. (terminate-and-stay-resident programs) (tutorial) (Cover Story)
Belling, Douglas
Microsoft Systems Journal, v7, n1, p41(19)
Jan-Feb, 1992
DOCUMENT TYPE: Cover Story ISSN: 0889-9932 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 6309 LINE COUNT: 00477

... save the current memory allocation configuration.
The addition of the SETVER command means the Get **Version Number** (**Function 30H**) call cannot be relied upon to return the true version of
DOS. This can...

7/3,K/36 (Item 36 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

01494230 SUPPLIER NUMBER: 11680198 (USE FORMAT 7 OR 9 FOR FULL TEXT)
What's the code? (designing a file finder utility) (Tech Section)
(Tutorial)
Stafford, David
Computer Shopper, v12, n1, p707(2)
Jan, 1992
DOCUMENT TYPE: Tutorial ISSN: 0886-0556 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1550 LINE COUNT: 00118

... source code. Let's look at it step by step.
main() first checks the DOS **version number** because the FullName()
function requires 3.0 or later. Next it checks Argc to verify that the
program received...

7/3,K/37 (Item 37 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01462103 SUPPLIER NUMBER: 11122947 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Expand batch file versatility with STRINGS.COM.
Boling, Douglas
PC Magazine, v10, n15, p419(9)
Sept 10, 1991
ISSN: 0888-8507 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 5044 LINE COUNT: 00379

... batch files in determining the system configuration: VER, ASK,
ENVFREE, ENVSIZE, and MASTERVAR.
The VER **function** returns a string containing the current DOS
version number. The major version number is reported in the hundreds
position, and the minor version numbers...

7/3,K/38 (Item 38 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01449939 SUPPLIER NUMBER: 11295365 (USE FORMAT 7 OR 9 FOR FULL TEXT)
A BASIC approach to programming in Windows, part I. (Lab Notes) (Microsoft
Windows graphical user interface) (includes related article on using
WordBasic in the Word for Windows working model) (tutorial)
Schulman, Andrew
PC Magazine, v10, n17, p393(9)
Oct 15, 1991
DOCUMENT TYPE: tutorial ISSN: 0888-8507 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 5633 LINE COUNT: 00429

... through the 3-pound Microsoft Windows Programmer's Reference, you
will eventually find the GetVersion() **function**. It returns a 2-byte value
with Windows' major **version number** (for example, 3) in the low byte and
the minor version or revision number (for...

7/3,K/39 (Item 39 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01441781 SUPPLIER NUMBER: 10991030 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Lanport-II solves comm server ills. (Hardware Review) (Microtest's
Lanport-II communications server; includes a related article on
Lanport-II's command interface) (evaluation)
Hurwicz, Mike
LAN Magazine, v6, n7, p126(8)

July, 1991

DOCUMENT TYPE: evaluation ISSN: 0898-0012 LANGUAGE: ENGLISH

RECORD TYPE: FULLTEXT; ABSTRACT

WORD COUNT: 4492 LINE COUNT: 00343

... enter the name of one of them.) Finally, you can list the Lanport-II firmware **version number**.

In addition, the "directory listing" **function** is more flexible when used from the command line: You can provide a file specification...

7/3,K/40 (Item 40 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

1418322 SUPPLIER NUMBER: 09805521 (USE FORMAT 7 OR 9 FOR FULL TEXT)

An introduction to the DOS Protected Mode Interface. (Power Programming)

(Column) (tutorial)

Duncan, Ray

PC Magazine, v10, n3, p365(5)

Feb 12, 1991

DOCUMENT TYPE: tutorial ISSN: 0888-8507 LANGUAGE: ENGLISH

RECORD TYPE: FULLTEXT; ABSTRACT

WORD COUNT: 2865 LINE COUNT: 00222

... implied mode switch.

* The Miscellaneous Services include address conversions, coprocessor management, debugging support, and a **function** to get the DPMI **version number**.

Nearly all of these DPMI **functions** are intended only for DOS extenders; they would ordinarily never be called directly by an...

7/3,K/41 (Item 41 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

01389886 SUPPLIER NUMBER: 09577619 (USE FORMAT 7 OR 9 FOR FULL TEXT)

One developer's appraisal of the New dBASE IV. (includes related article on user opinions of dBASE IV) (evaluation)

Buzzard, James

Data Based Advisor, v8, n11, p85(5)

Nov, 1990

DOCUMENT TYPE: evaluation ISSN: 0740-5200 LANGUAGE: ENGLISH

RECORD TYPE: FULLTEXT; ABSTRACT

WORD COUNT: 3689 LINE COUNT: 00285

... either check your dBASE directory for the date on DBASE.EXE or use the VERSION() **function** to get the "true" **version number** ("? VERSION(1)" returns "dBASE IV 1.1xx500 (07/17/90)" on my disks).

Faster installation...

7/3,K/42 (Item 42 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

01362778 SUPPLIER NUMBER: 08559220 (USE FORMAT 7 OR 9 FOR FULL TEXT)

The authorised version. (the right way to program the latest EMS specification version 4.0)

Clifford, D.A.

EXE, v5, n1, p24(5)

June, 1990

ISSN: 0268-6872 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT

WORD COUNT: 2391 LINE COUNT: 00180

... functions may be used where an EMS V3.0 handler is in operation (see Get **Version Number Function**), or where the interrupt handler is

able to restrict its own use of the page...

7/3,K/43 (Item 43 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01311325 SUPPLIER NUMBER: 07736544 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Getting inside PC viruses. (Vienna Virus, Jerusalem Virus, Cascade Virus
and New Zealand Virus) (includes related article on Brain and Italian
viruses)
Hirst, Joe
Tech PC User, v1, n9, p22(5)
May, 1989
ISSN: 0954-6995 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 4407 LINE COUNT: 00316

... positive response is received by a virus attached to a COM file
(and the returned version number is not less than three), another DOS
function (INT 21H, function DDH) will be called. This will cause the
resident version of the virus to move...

7/3,K/44 (Item 44 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01310147 SUPPLIER NUMBER: 07585718 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Getting the most from expanded memory with an EMS function library.
(Expanded Memory Specification Version 4.0)
Boling, Douglas
Microsoft Systems Journal, v4, n5, p15(19)
Sept, 1989
ISSN: 0889-9932 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 5715 LINE COUNT: 00438

... indicates the function was completed with no errors. The EMM
version is determined by calling function 7, Get Version. The version
number is returned as a B CD byte in AL.
The page frame segment can be...

7/3,K/45 (Item 45 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01305966 SUPPLIER NUMBER: 07421866 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Power programming. (expanded memory management)
Duncan, Ray
IBM Magazine, v8, n14, p379(5)
August, 1989
ISSN: 0888-8507 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1970 LINE COUNT: 00149

... to make sure the expanded memory hardware is present and working
properly. Next, the "get version number " function should be called;
this ensures that all of the EMS functions the program needs are...

7/3,K/46 (Item 46 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01300982 SUPPLIER NUMBER: 07388314 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Extended memory specification 2.x: taking advantage of the 80286 protected
mode. (includes related article on extended memory specification
functions)
Anderson, Chip

Microsoft Systems Journal, v4, n4, p17(10)

July, 1989

ISSN: 0889-9932

LANGUAGE: ENGLISH

RECORD TYPE: FULLTEXT; ABSTRACT

WORD COUNT: 4065

LINE COUNT: 00312

... functions. These functions will help give your programs instant access to more memory.

Get XMS **Version Number** (**Function** 00h) ARGs: AH = 00h RETs: AX = XMS **version number** BX = Driver internal revision number DX = 0001h if the HMA exists, 0000h otherwise

This function...

7/3,K/47 (Item 47 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

01297257 SUPPLIER NUMBER: 07596993 (USE FORMAT 7 OR 9 FOR FULL TEXT)

PM functions in non-PM programs. (Presentation Manager) (Environments)

(column)

Petzold, Charles

PC Magazine, v8, n11, p343(6)

June 13, 1989

DOCUMENT TYPE: column ISSN: 0888-8507

LANGUAGE: ENGLISH

RECORD TYPE: FULLTEXT; ABSTRACT

WORD COUNT: 2725 LINE COUNT: 00210

... program doesn't get loaded and so never even gets an opportunity to check the **version number**.

THE ALLOWABLE **FUNCTIONS**

Now that we have seen how some Presentation Manager functions can be used in a...

7/3,K/48 (Item 48 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

01289560 SUPPLIER NUMBER: 07126822 (USE FORMAT 7 OR 9 FOR FULL TEXT)

Drawing out DESQview power. (Quarterback Office Systems' DESQview

application program interface)

Hitt, Frederick J.

PC Tech Journal, v7, n4, p46(12)

April, 1989

ISSN: 0738-0194 LANGUAGE: ENGLISH

RECORD TYPE: FULLTEXT; ABSTRACT

WORD COUNT: 8352 LINE COUNT: 00676

... the function returns with an error code in AL and BX unchanged. Under DESQview, the **function** returns the DESQview **version number** in BX.

The SHADOW call asks DESQview for a window buffer address that the program...

7/3,K/49 (Item 49 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2004 The Gale Group. All rts. reserv.

01213709 SUPPLIER NUMBER: 06142820 (USE FORMAT 7 OR 9 FOR FULL TEXT)

Intro to OS-2 function calls. (OS-2 CALL-based API)

Petzold, Charles

PC Magazine, v6, n18, p375(6)

Oct 27, 1987

ISSN: 0888-8507 LANGUAGE: ENGLISH

RECORD TYPE: FULLTEXT

WORD COUNT: 2897 LINE COUNT: 00224

... version. The single parameter is a far pointer to a word that will receive the **version number**. You first indicate that this **function** is

external and requires a far call: EXTRN DOSGETVERSION:FAR In the data segment of...

...VERSION on the stack: PUSH DS PUSH OFFSET VERSION CALL DOSGETVERSION On return from the **function**, the high byte of VERSION will be the major **version number** and the low byte will be the minor version number. (The OS/2 Kernel included with the Microsoft OS/2 Software Development Kit returns a **version number** of 10.0.)

DOSBEEP and DOSGETVERSION are relatively simple **function** calls, of course. Many of the other OS/2 functions are more complex. DOSOPEN, for...

7/3,K/50 (Item 50 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01213483 SUPPLIER NUMBER: 06077254 (USE FORMAT 7 OR 9 FOR FULL TEXT)
OS-2 program entry conditions.
Petzold, Charles
PC Magazine, v6, n20, p371(5)
Nov 24, 1987
ISSN: 0888-8507 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2851 LINE COUNT: 00219

... and about the hardware of the computer. DOSGETVERSION, for example, will return the OS/2 **version number**.

You can use the DOSDEVCONFIG **function** to determine the hardware configuration of the computer, including the number of printers, RS-232...

7/3,K/51 (Item 51 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01212851 SUPPLIER NUMBER: 05158842 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Mouse software: see how they run. (mouse add-on utilities)
Prosise, Jeff
PC Magazine, v6, n13, p411(10)
July 21, 1987
ISSN: 0888-8507 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 6860 LINE COUNT: 00523

... to define default values for any and all registers and to revert to them instantly. **Function** FAh returns the RIL **version number** and whether or not the mouse driver is present.

The Programmer's Reference also outlines...

7/3,K/52 (Item 52 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01207940 SUPPLIER NUMBER: 04689661 (USE FORMAT 7 OR 9 FOR FULL TEXT)
WindowDOS. (Software Review) (DOS shells) (evaluation)
Pepper, Jon
PC Magazine, v6, n4, p218(2)
Feb 24, 1987
DOCUMENT TYPE: evaluation ISSN: 0888-8507 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 780 LINE COUNT: 00058

... will need them or like them. For example, Ctrl-E will pop up an Environment **function** that displays RAM usage, computer type, DOS **version number**, ROM date, and the memory load and interrupt vectors used by each program. This feature...

7/3,K/53 (Item 1 from file: 636)

DIALOG(R)File 636:Gale Group Newsletter DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

04635287 Supplier Number: 61592640 (USE FORMAT 7 FOR FULLTEXT)
News; Big screen in the Big Apple.
Bloomfield, Larry
Broadcast Engineering, pNA
March, 2000
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 3402

... area.

The Program Identifier executes a channel change based upon a program's episode and **version number**. Use of this **function** would permit a broadcaster to direct a viewer's attention to a broadcast of a...

7/3,K/54 (Item 2 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01187450 Supplier Number: 41096443 (USE FORMAT 7 FOR FULLTEXT)
Folio Corporation: On a Clear Day, the Views Are Very Fast
Electronic Services Update, pN/A
Jan, 1990
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 1229

... there is no denying that the true power of the system lies in the full- **function version**. No matter how thoroughly an infobase is menued, linked and otherwise prepared, the structure reflects the...

7/3,K/55 (Item 1 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)
(c) 2004 The Gale Group. All rts. reserv.

04645921 Supplier Number: 46835010 (USE FORMAT 7 FOR FULLTEXT)
Simulation forecasts complex flow streams from Ekofisk
The Oil and Gas Journal, p69
Oct 28, 1996
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 3716

... parameters between the propane and the heavier components, which were generated automatically by Hysim.

Calculator **function** limitations - Certain data such as the time and date and **version number** of Hysim are not directly available to the calculator. A dummy file needed to be...

7/3,K/56 (Item 1 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

10254891 SUPPLIER NUMBER: 20789306 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Act! Gets Upgrade -- PIM Is Useful, But Big Changes Are Few. (Symantec's Act! 4.0 personal information manager) (Software Review) (Evaluation)
Feibus, Andy
InformationWeek, n685, p118(1)
June 8, 1998
DOCUMENT TYPE: Evaluation ISSN: 8750-6874 LANGUAGE: English
RECORD TYPE: Fulltext
WORD COUNT: 1189 LINE COUNT: 00093

... Act! to eliminate the need for a bolted-on product like SideAct!
Although the added **function** in this product makes me think its **version number** should be 3.1 and not 4.0, Symantec's Act! is still an excellent...

7/3,K/57 (Item 2 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

09137185 SUPPLIER NUMBER: 18822943 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Simulation forecasts complex flow streams from Ekofisk. (Phillips Petroleum Co. Norway A/S production management software)
Arnes, Fatos Cecen; Lillejord, Henning; Vieler, Andrew
Oil and Gas Journal, v94, n44, p69(7)
Oct 28, 1996
ISSN: 0030-1388 LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 4135 LINE COUNT: 00345

... parameters between the propane and the heavier components, which were generated automatically by Hysim.
* Calculator **function** limitations--Certain data such as the time and date and **version number** of Hysim are not directly available to the calculator. A dummy file needed to be...

7/3,K/58 (Item 3 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

02029046 SUPPLIER NUMBER: 03291111 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Wide screen computing. (evaluation)
Blank, George
Creative Computing, v10, p26(7)
June, 1984
DOCUMENT TYPE: evaluation ISSN: 0097-8140 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 4383 LINE COUNT: 00324

... the tab positions indicated for laying out your text. Also listed are the program name, **version number**, and the same of your file. The screen **functions** as a moving window, so the document can be wider and much longer than that...

7/3,K/59 (Item 1 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)
(c) 2004 ProQuest Info&Learning. All rts. reserv.

01649743 03-00733
Act! Gets upgrade
Feibus, Andy
Informationweek n685 PP: 118-120 Jun 8, 1998
ISSN: 8750-6874 JRNL CODE: IWK
WORD COUNT: 1139

...TEXT: Act! to eliminate the need for a bolted-on product like SideAct!

Although the added **function** in this product makes me think its **version number** should be 3.1 and not 4.0, Symantec's Act! is still an excellent
...

7/3,K/60 (Item 2 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)
(c) 2004 ProQuest Info&Learning. All rts. reserv.

00552674 91-27031

DR DOS 5.0 Adds Value to Compete with the Leading Brand

Glass, Brett

InfoWorld v13n21 PP: 91-93 May 27, 1991

ISSN: 0199-6649 JRNL CODE: IFW

WORD COUNT: 2754

...TEXT: you enter the VER command, the system displays "DR DOS 5.0."
However, the internal **function** (30h) that tells applications the **version number** of DOS, reports Version 3.31.

This white lie actually has some advantages. If you...

7/3,K/61 (Item 1 from file: 647)

DIALOG(R)File 647:CMP Computer Fulltext

(c) 2004 CMP Media, LLC. All rts. reserv.

01163777 CMP ACCESSION NUMBER: IWK19980608S0054

Act! Gets Upgrade - PIM Is Useful, But Big Changes Are Few

Andy Feibus

INFORMATIONWEEK, 1998, n 685, PG118

PUBLICATION DATE: 980608

JOURNAL CODE: IWK LANGUAGE: English

RECORD TYPE: Fulltext

SECTION HEADING: Software

WORD COUNT: 1118

... Act! to eliminate the need for a bolted-on product like SideAct!

Although the added **function** in this product makes me think its **version number** should be 3.1 and not 4.0, Symantec's Act! is still an excellent...

7/3,K/62 (Item 2 from file: 647)

DIALOG(R)File 647:CMP Computer Fulltext

(c) 2004 CMP Media, LLC. All rts. reserv.

01149252 CMP ACCESSION NUMBER: WIN19980101S0110

Windows DNA? Been There - WinTune 98 used the hottest development architecture-DNA-before it even had a name. (Web Dev)

Martin Heller

WINDOWS MAGAZINE, 1998, n 901, PG291

PUBLICATION DATE: 980101

JOURNAL CODE: WIN LANGUAGE: English

RECORD TYPE: Fulltext

SECTION HEADING: How To

WORD COUNT: 1468

... precision, but I couldn't stand it. Eventually, I discovered that VBScript does report its **version number**, and that I could conditionally execute **functions** safely. That led me to write Function R2(a), which rounds smallish numbers to two...

File 8: Ei Compendex(R) 1970-2004/Apr W4
 (c) 2004 Elsevier Eng. Info. Inc.
 File 35: Dissertation Abs Online 1861-2004/Apr
 (c) 2004 ProQuest Info&Learning
 File 65: Inside Conferences 1993-2004/Apr W4
 (c) 2004 BLDSC all rts. reserv.
 File 2: INSPEC 1969-2004/Apr W4
 (c) 2004 Institution of Electrical Engineers
 File 94: JICST-EPlus 1985-2004/Apr W2
 (c) 2004 Japan Science and Tech Corp(JST)
 File 483: Newspaper Abs Daily 1986-2004/May 01
 (c) 2004 ProQuest Info&Learning
 File 6: NTIS 1964-2004/May W1
 (c) 2004 NTIS, Intl Cpyrght All Rights Res
 File 144: Pascal 1973-2004/Apr W4
 (c) 2004 INIST/CNRS
 File 434: SciSearch(R) Cited Ref Sci 1974-1989/Dec
 (c) 1998 Inst for Sci Info
 File 34: SciSearch(R) Cited Ref Sci 1990-2004/Apr W4
 (c) 2004 Inst for Sci Info
 File 99: Wilson Appl. Sci & Tech Abs 1983-2004/Mar
 (c) 2004 The HW Wilson Co.
 File 583: Gale Group Globalbase(TM) 1986-2002/Dec 13
 (c) 2002 The Gale Group
 File 266: FEDRIP 2004/Mar
 Comp & dist by NTIS, Intl Copyright All Rights Res
 File 95: TEME-Technology & Management 1989-2004/Apr W2
 (c) 2004 FIZ TECHNIK
 File 438: Library Lit. & Info. Science 1984-2004/Mar
 (c) 2004 The HW Wilson Co

Set	Items	Description
S1	180	VERSION() (NUMBER OR NO OR ID)
S2	4	S1(10N)FUNCTION
S3	3	RD (unique items)

3/5/1 (Item 1 from file: 8)
DIALOG(R)File 8:EI Compendex(R)
(c) 2004 Elsevier Eng. Info. Inc. All rts. reserv.

00760647 E.I. Monthly No: EI7811084742 E.I. Yearly No: EI78069599
Title: DIFFERENTIAL PULSE-CODE MODULATION OF THE WIENER PROCESS.
Author: Hayashi, Akira
Corporate Source: Kanazawa Inst of Technol, Ishikawa-ken, Jpn
Source: IEEE Transactions on Communications v COM-26 n 6 Jun 1978 p
881-887

Publication Year: 1978
CODEN: IECMBT ISSN: 0096-1965
Language: ENGLISH
Journal Announcement: 7811

Abstract: The performance of DPCM with uniform quantization with the Wiener process input is analyzed. The approach taken is to establish an equation for the characteristic **function** of error distribution and then to solve its steady-state **version**. No use is made in the derivation of the approximating concepts of slope overload error and granular error. 9 refs.

Descriptors: *PULSE CODE MODULATION; CODES, SYMBOLIC--Error Statistics

Identifiers: DIFFERENTIAL PULSE CODE MODULATION; WIENER PROCESS

Classification Codes:

716 (Radar, Radio & TV Electronic Equipment); 723 (Computer Software);
922 (Statistical Methods)

71 (ELECTRONICS & COMMUNICATIONS); 72 (COMPUTERS & DATA PROCESSING); 92
(ENGINEERING MATHEMATICS)

3/5/2 (Item 1 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 2004 Institution of Electrical Engineers. All rts. reserv.

5981369

Title: A marginally sharpened Act! [personal information manager]

Author(s): Feibus, A.

Journal: InformationWeek no.32 p.83

Publisher: Emap Computing & CMP Media Inc,

Publication Date: 24 June-7 July 1998 Country of Publication: UK

CODEN: INFWF5

Material Identity Number: G220-98013

Language: English Document Type: Journal Paper (JP)

Treatment: Practical (P); Product Review (R)

Abstract: Although the added **function** in this product suggests its **version number** should be 3.1 and not 4.0, Symantec's Act! is still an excellent PIM for those users who are looking for a tool that has more power and better integration than more widely used tools such as Microsoft Outlook or Lotus Organizer. When it comes to choosing a PIM to maintain personal phone lists, calendar of appointments, and to-do lists, it is worth considering Act! 4.0. (0 Refs)

Subfile: D

Descriptors: software reviews; time management

Identifiers: Symantec Act! 4.0; personal information manager

Class Codes: D2010 (Business and professional)

Copyright 1998, IEE

3/5/3 (Item 2 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 2004 Institution of Electrical Engineers. All rts. reserv.

02639509 INSPEC Abstract Number: C86024940

Title: Calling MS-DOS from Turbo Pascal-it's a breeze

Author(s): Swan, T.

Journal: Programmer's Journal vol.3, no.4 p.30-2

Publication Date: July-Aug. 1985 Country of Publication: USA

CODEN: PRGJE7 ISSN: 0747-5861

Language: English Document Type: Journal Paper (JP)

Treatment: Practical (P)

Abstract: The built-in MSDOS procedure in Turbo Pascal makes calling operating system routines a breeze. One can read the system date, check the DOS **version number**, change directory paths, list directories, or call any other DOS **function** -without once having to drop into assembly language. (0 Refs)

4/5/1 (Item 1 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

009485 **Image available**
PROGRAM DOWNLOAD SYSTEM, BROADCASTING COMMUNICATION FUSION TERMINAL AND
PROGRAM DOWNLOAD METHOD

PUB. NO.: 2003-223387 [JP 2003223387 A]
PUBLISHED: August 08, 2003 (20030808)
INVENTOR(s): FUNAYA KOICHI
CHISHIMA HIROSHI
SATO NAOKI
KATO AKIRA
KANEDA SATORU
SHIBATA SHUICHI
APPLICANT(s): NEC CORP
APPL. NO.: 2002-019929 [JP 200219929]
FILED: January 29, 2002 (20020129)
INTL CLASS: G06F-013/00; G06F-009/445; H04N-005/44; H04N-007/173

ABSTRACT

PROBLEM TO BE SOLVED: To simplify and quicken the download processing of the browser **software** of a broadcasting communication fusion terminal.
SOLUTION: A module providing server 3 preliminarily stores a data processing module for realizing a part of the **function** of a data processing program and its **version number**. A broadcasting station device 1 broadcasts download announcement information including the address of the module providing server 3 in which the latest data processing module and the version number are stored, together with data broadcasting. A terminal 5 receives the download announcement information together with the data broadcasting, and separates them and analyzes the separation result. The terminal 5 requests the data processing module of the latest version number to a module providing server 3 when the latest version number is newer than the version number which is being used at present. The module providing server 3 downloads the requested data processing module to the terminal 5 in response to the request from the terminal 5.

COPYRIGHT: (C)2003,JPO

4/5/2 (Item 2 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

07435802 **Image available**
FILE MANAGEMENT METHOD AND PROGRAM THEREFOR, AND RECORDING MEDIUM

PUB. NO.: 2002-304312 [JP 2002304312 A]
PUBLISHED: October 18, 2002 (20021018)
INVENTOR(s): INOUE TAKESHI
APPLICANT(s): MATSUSHITA ELECTRIC WORKS LTD
APPL. NO.: 2001-107673 [JP 2001107673]
FILED: April 05, 2001 (20010405)
INTL CLASS: G06F-012/00; G06F-009/44

ABSTRACT

PROBLEM TO BE SOLVED: To improve efficiency of file management by grouping and managing a plurality of files together with **version number** information with a conventional file **version number** management **function** as a base.

SOLUTION: When updating an individual file of the plurality of files constituting a **software** or the like in a process for creating, developing or modifying the **software** or the like, the version number information is added to the file, and the plurality of files are grouped and managed together with the version number information. By storing the plurality of files added with the version number information into different folders by

File 347:JAPIO Nov 1976-2003/Dec(Updated 040402)

(c) 2004 JPO & JAPIO

File 350:Derwent WPIX 1963-2004/UD,UM &UP=200427

(c) 2004 Thomson Derwent

Set	Items	Description
S1	590	VERSION() (NUMBER OR NO OR ID)
S2	23	S1(10N)FUNCTION
S3	1164064	CODE OR PROGRAMMING()LANGUAGE? ? OR OOPL OR SOFTWARE OR AP- PLICATION? ? OR LIBRAR???
S4	121	S2 AND S3

kinds, and specifying and extracting arbitrary version number information of an arbitrary file from the plurality of folders, a combination of the plurality of files is managed as one group. Information wherein group information for managing the plurality of files added with the version number information as the one group is plurally combined is managed as high-order hierarchical group information.

COPYRIGHT: (C)2002,JPO

4/5/3 (Item 3 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

07390417 **Image available**
SUPPORT TOOL FOR CONTROLLER

PUB. NO.: 2002-258918 [JP 2002258918 A]
PUBLISHED: September 13, 2002 (20020913)
INVENTOR(s): WADA HIROYUKI
APPLICANT(s): FUJI ELECTRIC CO LTD
APPL. NO.: 2001-052684 [JP 200152684]
FILED: February 27, 2001 (20010227)
INTL CLASS: G05B-019/05; G06F-009/45

ABSTRACT

PROBLEM TO BE SOLVED: To improve reliability by increasing the function of supporting a controller.

SOLUTION: A support tool 2 for supporting a controller 1 is provided with a function for managing assemblers in all generations for every version and calling and executing the designated version of assembler, and a **function** for storing the **version number** (version **code**) of the assembler assembled for the source program of the controller in a memory 3. Thus, it is possible to simplify maintenance and to improve reliability by reducing a state that machine word objects are different from each other in the controller 1 and the support tool 2 as much as possible.

COPYRIGHT: (C)2002,JPO

4/5/4 (Item 4 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

06527046 **Image available**
SOFTWARE DISTRIBUTION SYSTEM

PUB. NO.: 2000-112766 [JP 2000112766 A]
PUBLISHED: April 21, 2000 (20000421)
INVENTOR(s): IKEDA SUSUMU
APPLICANT(s): NEC CORP
APPL. NO.: 10-283702 [JP 98283702]
FILED: October 06, 1998 (19981006)
INTL CLASS: G06F-009/445; G06F-009/06; G06F-013/00

ABSTRACT

PROBLEM TO BE SOLVED: To provide a **software** distribution system integrally managing the file distribution of a client/server **application**.

SOLUTION: In distributing a file to a personal computer(PC) client machine 104 from the manager function 105 of a **software** distribution server machine 101, the **function** 105 registers distribution file information and the **version number** of another **application** number necessary for the operation of the distribution file and transmits the distribution file to the agent function 106 of a PC client machine 104. The function 106 receives a file and a command and returns an executed result to the function 105. When the executed result is normal, the function 105 updates

information of a present version.

COPYRIGHT: (C)2000,JPO

4/5/5 (Item 5 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

05860307 **Image available**
INFORMATION PROVIDING SYSTEM

PUB. NO.: 10-143407 [JP 10143407 A]
PUBLISHED: May 29, 1998 (19980529)
INVENTOR(s): MATSUMURA TAKAHIRO
TAKANO MASAJI
SUGIMURA TOSHIAKI
KATAGIRI MASAJI
APPLICANT(s): NIPPON TELEGR & TELEPH CORP <NTT> [000422] (A Japanese
Company or Corporation), JP (Japan)
APPL. NO.: 08-313060 [JP 96313060]
FILED: November 08, 1996 (19961108)
INTL CLASS: [6] G06F-012/00; G06F-012/00; G06F-013/00; G06F-013/00;
G06F-015/00
JAPIO CLASS: 45.2 (INFORMATION PROCESSING -- Memory Units); 45.4
(INFORMATION PROCESSING -- Computer **Applications**)

ABSTRACT

PROBLEM TO BE SOLVED: To provide an information providing system with which it is not necessary to pay a communication tariff more than required when the information file to be downloaded does not exist and further, average time from the update of information held in server equipment to the download is shortened. .

CONCLUSION: Server equipment SB has a data base DB recording call originating subaddress information or the like corresponding to communication lines 11 and 12 to which respective pieces of terminal equipment T1-T3 are connected, information held at the respective pieces of terminal equipment T1-T3 and the version numbers of that information. In this case, when an incoming call from any one of terminal equipment T1-T3 is detected, based on the call originating subaddress information or the like reported from a communication network NW having a call originating subaddress information reporting function, etc., the version number of information held at that terminal equipment is retrieved from the data base DB and only when it is confirmed that this retrieved version is not the latest one, a response is applied. Then, the file of difference between the information of the latest version and the information of the terminal equipment is downloaded.

4/5/6 (Item 6 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

05608370 **Image available**
PROCESS OPERATION FLOW MANAGING SYSTEM

PUB. NO.: 09-223170 [JP 9223170 A]
PUBLISHED: August 26, 1997 (19970826)
INVENTOR(s): HIROYA MITSURU
EGAWA YASUHIRO
APPLICANT(s): SONY CORP [000218] (A Japanese Company or Corporation), JP
(Japan)
APPL. NO.: 08-053928 [JP 9653928]
FILED: February 16, 1996 (19960216)
INTL CLASS: [6] G06F-017/60
JAPIO CLASS: 45.4 (INFORMATION PROCESSING -- Computer **Applications**)

ABSTRACT

PROBLEM TO BE SOLVED: To prepare a process operation flow derived from a partial change by changing only a corresponding part in a basic process operation flow.

SOLUTION: A **version number** issue managing **function** part 20 issues version numbers to process operation flows to be prepared in accordance with preparation order to constitute respective prepared process operation flows hierarchically. A process flow managing function part 30 newly prepares only an independent part in a process operation flow to be prepared so that a common part in an upper process operation flow can be used for the corresponding part of the prepared process operation flow and stores the prepared process operation flow in a data base in relation with a received version number. In the case of retrieving a prescribed process operation flow, version numbers are hierarchically traced to the upper side to collect the corresponding process operation flow.

4/5/7 (Item 7 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

04352416 **Image available**
SATELLITE DATA BROADCASTING SYSTEM

PUB. NO.: 05-344116 [JP 5344116 A]
PUBLISHED: December 24, 1993 (19931224)
INVENTOR(s): UENO NOBUO
APPLICANT(s): FUJITSU LTD [000522] (A Japanese Company or Corporation), JP
(Japan)
APPL. NO.: 04-151032 [JP 92151032]
FILED: June 11, 1992 (19920611)
INTL CLASS: [5] H04L-009/06; H04L-009/14; H04B-007/212; H04L-012/18
JAPIO CLASS: 44.3 (COMMUNICATION -- Telegraphy); 34.4 (SPACE DEVELOPMENT
-- Communication); 44.2 (COMMUNICATION -- Transmission
Systems)
JOURNAL: Section: E, Section No. 1530, Vol. 18, No. 174, Pg. 145,
March 24, 1994 (19940324)

ABSTRACT

PURPOSE: To detect the noncoincidence of keys at a receiving station without decreasing the packet transmission capacity of transmitting information concerning the satellite data broadcasting system for ciphering and transmitting the transmitting information.

CONSTITUTION: In the satellite data broadcasting system provided with a center station A and a receiving station B, the center station is provided with a key version number generating means 4 to generate and transmit the number of versions for network keys, key information generating means 2 to generate key information by ciphering the network keys and adding the **version number** of key, and key edition number fetching **function** 31 added to a frame control **code** generating section 3. The receiving station is provided with a key information receiving means 6 to separate the key version number from the inputted key information and to store it, and key version number collating means 7 to collate the key information receiving means with the respective key version number extracted at a frame control **code** extracting section 5 and to output the collated result.

4/5/8 (Item 8 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

03411824 **Image available**
INSTALLATION INSPECTING DEVICE

PUB. NO.: 03-074724 [JP 3074724 A]
PUBLISHED: March 29, 1991 (19910329)
INVENTOR(s): TAKECHI YASUhide

APPLICANT(s): NEC CORP [000423] (A Japanese Company or Corporation), JP
(Japan)
APPL. NO.: 01-210690 [JP 89210690]
FILED: August 17, 1989 (19890817)
INTL CLASS: [5] G06F-009/06
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 1217, Vol. 15, No. 242, Pg. 9, June
21, 1991 (19910621)

ABSTRACT

PURPOSE: To reduce the load of a user required for an inspecting job and at the same time to prevent the misinspection by checking whether a program or a file is completely registered or not based on the information set previously.

CONSTITUTION: A program file 2 is provided to store the information set previously on a program together with a data file 3 which stores the information set previously and necessary for actuation of the program. An installation inspecting means 1 inspects whether the program is completely registered or not into a program **library** 6 and a file name information file 7 based on the contents of both files 2 and 3 which are loaded to an electronic computer 5 with an installation inspecting instruction of a user. At the same time, the means 1 inspects whether a system program has a **function** necessary for the program action or not based on the **version number** (updating recording frequency). Thus it is possible to reduce load of the user required for an inspecting job and also to avoid the inconvenience of the program action caused by the misinspection.

4/5/9 (Item 9 from file: 347)

DIALOG(R)File 347:JAPIO

(c) 2004 JPO & JAPIO. All rts. reserv.

02819962 **Image available**

SOFTWARE REVISION SYSTEM FOR FACSIMILE EQUIPMENT

PUB. NO.: 01-117562 [JP 1117562 A]
PUBLISHED: May 10, 1989 (19890510)
INVENTOR(s): KUMAGAI YASUHIRO
APPLICANT(s): NEC CORP [000423] (A Japanese Company or Corporation), JP
(Japan)
APPL. NO.: 62-275171 [JP 87275171]
FILED: October 30, 1987 (19871030)
INTL CLASS: [4] H04N-001/00; G06F-009/06; H04N-001/32
JAPIO CLASS: 44.7 (COMMUNICATION -- Facsimile); 45.1 (INFORMATION
PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: E, Section No. 804, Vol. 13, No. 359, Pg. 39, August
10, 1989 (19890810)

ABSTRACT

PURPOSE: To always up-to-date the **version number** of the **software** of a facsimile equipment by providing a **function** polling a center equipment so as to check the **version number** of the **software** from a facsimile equipment at the terminal equipment side.

CONSTITUTION: A control circuit 6 applies polling to a center equipment 10 via a communication control circuit 5 and a communication line 11 at each prescribed period set in advance or at **application** of daily power supply or according to the command of the operator. The center equipment 10 sends the latest version number information of the **software** in use for the facsimile equipment to be managed. A comparator 4 compares the version number information of the **software** of its own equipment stored in a memory 2 with the version number information sent from the center equipment 10 of the memory 3 and when they are dissident, the comparator 4 requests down-loading of the newest **software** to its own equipment to the center equipment 10 via the communication control circuit 5.

4/5/10 (Item 1 from file: 350)

DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

015632030 **Image available**
WPI Acc No: 2003-694212/200366
XRPX Acc No: N03-554903

User terminal has control function unit which functions to setup,
register and change internet protocol version ID corresponding to
user terminal, in response to receiving services from various service
provision firms

Patent Assignee: NIPPON DENSHI SERVICE KK (NIDE-N)

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
JP 2003264578	A	20030919	JP 200264686	A	20020311	200366 B

Priority Applications (No Type Date): JP 200264686 A 20020311

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
JP 2003264578	A	6	H04L-012/56	

Abstract (Basic): JP 2003264578 A

NOVELTY - A control function unit (12) functions to setup, register
and change internet protocol version 6 (IPv6) identifier (ID)
corresponding to the user terminal (1), in response to receiving
services from various service provision firms.

DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for
computer **software** for operating user terminal.

USE - User terminal used for receiving various services provided by
service provision firms such as banks, credit firms, travel agencies,
ticket agent firms and hotels.

ADVANTAGE - Enables rapidly switching between different provision
services by appropriately changing the IPv6 ID.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of
the user terminal. (Drawing includes non-English language text).

user terminal (1)
terminal function unit (11)
internet protocol version (IPv6) control function unit (12)
service controller (14)
input output controller (15)
pp; 6 DwgNo 2/4

Title Terms: USER; TERMINAL; CONTROL; FUNCTION; UNIT; FUNCTION; REGISTER;
CHANGE; PROTOCOL; VERSION; ID; CORRESPOND; USER; TERMINAL; RESPOND;
RECEIVE; SERVICE; VARIOUS; SERVICE; PROVISION

Derwent Class: T01; W01

International Patent Class (Main): H04L-012/56

International Patent Class (Additional): G06F-015/00; H04L-029/06

File Segment: EPI

4/5/11 (Item 2 from file: 350)

DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

014527792 **Image available**
WPI Acc No: 2002-348495/200238

Method for developing and managing project through project schedule board
interlocked with source code version management

Patent Assignee: KOREA TELECOM (KOTE-N)

Inventor: BYUN G S; KIM J Y; LEE J H

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
KR 2001064882	A	20010711	KR 9959261	A	19991220	200238 B

Priority Applications (No Type Date): KR 9959261 A 19991220

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
-----------	------	--------	----------	--------------

Abstract (Basic): KR 2001064882 A

NOVELTY - A method for developing and managing a project through a project schedule board interlocked with a management of a source code version is provided to unitedly manage a management of a source code version and a project development by interlocking a result of the management of the source code version with a web site of the management of the project development in a source code and project integrated development management system.

DETAILED DESCRIPTION - A promoting schedule of a project is prepared (201). A plan for embodying a detail function is prepared (202). A completion request function item list is prepared by development stages. In addition, A charged developer list is prepared by function items (203). A completion list is prepared by development stages (204). A version number is given by source code files. In addition, a related function item list is prepared (205). It is judged whether a point of present time is a closing time of a project stage proceeded (206). It is judged whether there's a new change request in a specific source code file (207). It is judged whether a change of the specific source file is needed (208). It is judged whether a specific function item is completed (209). A project server accumulates and calculates an accomplishment rate of personal aim by stages and so on (210). Information of a project schedule board is corrected and displayed (211).

pp; 1 DwgNo 1/10

Title Terms: METHOD; DEVELOP; MANAGE; PROJECT; THROUGH; PROJECT; SCHEDULE; BOARD; INTERLOCKING; SOURCE; CODE ; VERSION; MANAGEMENT

Derwent Class: T01

International Patent Class (Main): G06F-017/60

File Segment: EPI

4/5/12 (Item 3 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

012707756 **Image available**

WPI Acc No: 1999-513865/199943

XRPX Acc No: N99-383504

Mask pattern check procedure for LSI - involves comparing version number of mask pattern, with newest version number of corresponding function cell

Patent Assignee: RICOH KK (RICO)

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
JP 11224278	A	19990817	JP 9827060	A	19980209	199943 B

Priority Applications (No Type Date): JP 9827060 A 19980209

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 11224278	A		5	G06F-017/50	

Abstract (Basic): JP 11224278 A

NOVELTY - The version number of the mask pattern of the semiconductor circuit, is compared with the newest version number of the corresponding function cell.

USE - For LSI.

ADVANTAGE - The check is made at high speed. The trouble and the reason for the alteration in the old version is displayed. So the problem and the replacement is found and performed. The mask pattern data which does not exist in the library is found, since the check is made for each function cell. The check is performed reliably and processing time is also less. DESCRIPTION OF DRAWING(S) - The figure represents mask pattern check apparatus.

File 348:EUROPEAN PATENTS 1978-2004/Apr W04

(c) 2004 European Patent Office

File 349:PCT FULLTEXT 1979-2002/UB=20040415,UT=20040408

(c) 2004 WIPO/Univentio

Set	Items	Description
S1	3056	VERSION() (NUMBER OR NO OR ID)
S2	115	S1(10N)FUNCTION
S3	39	S2(100N) (CODE OR PROGRAMMING()LANGUAGE? ? OR OOPL OR SOFTW- ARE OR APPLICATION? ? OR LIBRAR???)
S4	23\	S3 AND IC=G06F

4/3,K/7 (Item 7 from file: 348)
DIALOG(R) File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

00677970

SYSTEM FOR OBJECT ORIENTED DYNAMIC LINKING BASED UPON A CATALOG OF
REGISTERED FUNCTION SET OR CLASS IDENTIFIERS
OBJEKTORIENTIERTES DYNAMISCHES "LINK"-SYSTEM, WELCHES AUF KATALOGISIERTE
FUNKTIONEN UND KLASSEN ZUGREIFT

SYSTEME DE LIAISON DYNAMIQUE ORIENTEE OBJETS SUR LA BASE D'UN CATALOGUE
D'IDENTIFICATEURS ENREGISTRES DE JEUX OU DE CLASSES DE FONCTIONS

PATENT ASSIGNEE:

APPLE COMPUTER, INC., (1211950), 20525 Mariani Avenue, Cupertino,
California 95014, (US), (applicant designated states: DE;FR;GB;IT)

INVENTOR:

COWSAR, George, Curtis, 4598 Winding Way, San Jose, CA 95129, (US)
PLUMMER, Christopher, Jordan, 903 Erie Circle, Milpitas, CA 95035, (US)
QUINN, Michael, John, 919 Hedegard Avenue, Campbell, CA 95008-1810, (US)

LEGAL REPRESENTATIVE:

Tomlinson, Kerry John (36771), Frank B. Dehn & Co., European Patent
Attorneys, 179 Queen Victoria Street, London EC4V 4EL, (GB)

PATENT (CC, No, Kind, Date): EP 706684 A1 960417 (Basic)
EP 706684 B1 981104
WO 9501598 950112

APPLICATION (CC, No, Date): EP 94922068 940630; WO 94US7424 940630

PRIORITY (CC, No, Date): US 85187 930630

DESIGNATED STATES: DE; FR; GB; IT

INTERNATIONAL PATENT CLASS: G06F-009/44 ; G06F-009/445

NOTE:

No A-document published by EPO

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	9845	1677
CLAIMS B	(German)	9845	1505
CLAIMS B	(French)	9845	1904
SPEC B	(English)	9845	18404
Total word count - document A			0
Total word count - document B			23490
Total word count - documents A + B			23490

INTERNATIONAL PATENT CLASS: G06F-009/44 ...

... G06F-009/445

...SPECIFICATION number fields are used to insure that the cached
information is still valid (if the **library** unloaded and a new version
of a class or function set in the **library** was dragged into a registered
folder, any cached information is invalid). The version number fields
contain information about the **version number** of the class or
function set that your **application** (**code** resource, extension, etc.)
linked with, and the fClassIDStr field contains the actual ClassID of the
...of object-oriented ability for ordinary functions.

version =

This declaration defines the version of the **function** set. The
version number is in the standard Apple **version number** form:
#.#(.#). The version number may not have the extra release information
(like b2) on it...

...numbers separated either by 3 dots (...) or an ellipsis character
(option-;). This indicates the minimum **version number** of the
function set that this **function** set is backwards-compatible with, and
the current **version number** of the **function** set. Nothing is done
with this information in version 1.0 of SLM, but future...

...information. If you do not specify a version number, the version number
specified in the " **Library** " declaration will be assumed. This may be a

#defined symbol.

exports =
This declares a comma...

...CLAIMS set of the particular member function.

9. The apparatus of claim 8, further comprising
a **library** (203-223) of function sets in the memory, function sets in the **library** having function set identifiers, and including member functions; and
wherein the function sets are assigned...

...a standard protocol, and the dispatch record further includes version information linked with the client **application** indicating a minimum version number (221) supported by the client **application** for the **function** set of the particular member **function** ;
the **library** includes a **version number** (214) for the corresponding **function** sets, and
the link engine includes **code** responsive to the version information in the dispatch record and the version number in the **library** to insure that the client **application** supports a version of the function set of the particular member function.
10. The apparatus of claim 1, further comprising:
a **library** (203-223) of function sets stored in the memory, function sets in the **library** having function set identifiers, and including member functions; and wherein the link engine includes:
a...

4/3,K/8 (Item 8 from file: 348)

DIALOG(R) File 348:EUROPEAN PATENTS

(c) 2004 European Patent Office. All rts. reserv.

00652104

DISTRIBUTED WORK FLOW MANAGEMENT

VERTEILTES ARBEITSFLUSS-MANAGEMENT

GESTION REPARTIE DU DEROULEMENT DES OPERATIONS

PATENT ASSIGNEE:

Van-Ka BV, (3059760), Parnassustoren, Locatellikade 1, 1076 AZ AMSTERDAM,
(NL), (Proprietor designated states: all)

INVENTOR:

Nauckhoff, Sven, Sibyllegatan 43-45, 144 42 Stockholm, (SE)

LEGAL REPRESENTATIVE:

Akerman, Marten Lennart et al (69671), Albihns Malmo AB Box 4289, 203 14
Malmo, (SE)

PATENT (CC, No, Kind, Date): EP 788631 A1 970813 (Basic)
EP 788631 B1 020130
WO 9420910 940915

APPLICATION (CC, No, Date): EP 94909385 940301; WO 94SE173 940301

PRIORITY (CC, No, Date): SE 93671 930301

DESIGNATED STATES: DE; FR; GB; SE

INTERNATIONAL PATENT CLASS: **G06F-017/60**

NOTE:

No A-document published by EPO

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	200205	4107
CLAIMS B	(German)	200205	3720
CLAIMS B	(French)	200205	5081
SPEC B	(English)	200205	23619
Total word count - document A			0
Total word count - document B			36527
Total word count - documents A + B			36527

INTERNATIONAL PATENT CLASS: **G06F-017/60**

...SPECIFICATION Some implementation notes:

1. The DEACTIVATE(underscore)CASE is performed by the ATX when the **Application** Task has terminated.

If the case has been created or changed, then the **function** stores the new case **version number** in the case working control and data pools and closes them. It "commits" the new...

4/3,K/9 (Item 9 from file: 348)

DIALOG(R) File 348:EUROPEAN PATENTS

(c) 2004 European Patent Office. All rts. reserv.

00324410

Dynamically adaptive environment for computer programs.

Dynamische Umgebungsanpassung von Rechnerprogrammen.

Adaptation dynamique de l'environnement de programmes de calculateurs.

PATENT ASSIGNEE:

International Business Machines Corporation, (200120), Old Orchard Road,
Armonk, N.Y. 10504, (US), (applicant designated states:
BE;CH;DE;ES;FR;GB;IT;LI;NL;SE)

INVENTOR:

Evans, Wayne O'Neil, 3104 14th Avenue Northwest, Rochester Minnesota
55901, (US)

LEGAL REPRESENTATIVE:

Lattard, Nicole et al (16571), Compagnie IBM France Departement de
Propriete Intellectuelle, F-06610 La Gaude, (FR)

PATENT (CC, No, Kind, Date): EP 317478 A2 890524 (Basic)

EP 317478 A3 910911

EP 317478 B1 950517

APPLICATION (CC, No, Date): EP 88480058 881011;

PRIORITY (CC, No, Date): US 121965 871117

DESIGNATED STATES: BE; CH; DE; ES; FR; GB; IT; LI; NL; SE

INTERNATIONAL PATENT CLASS: **G06F-009/44**

ABSTRACT WORD COUNT: 207

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
----------------	----------	--------	------------

CLAIMS A	(English)	EPABF1	414
----------	-----------	--------	-----

CLAIMS B	(English)	EPAB95	448
----------	-----------	--------	-----

CLAIMS B	(German)	EPAB95	407
----------	----------	--------	-----

CLAIMS B	(French)	EPAB95	485
----------	----------	--------	-----

SPEC A	(English)	EPABF1	6586
--------	-----------	--------	------

SPEC B	(English)	EPAB95	6747
--------	-----------	--------	------

Total word count - document A	7000
-------------------------------	------

Total word count - document B	8087
-------------------------------	------

Total word count - documents A + B	15087
------------------------------------	-------

INTERNATIONAL PATENT CLASS: **G06F-009/44**

...SPECIFICATION one list or a plurality of lists with each list having a different sequence of **libraries**. The list could be selected based on the version ID.

For example, a command indicated...

...step 132 could use search routine 144 to search libraries according the version priority -- V1 **library**, V"N" **library**, User **library** 1, User **library** 2, etc. This priority or sequence of listed **libraries** could be different for each version and could be selected as a **function** of **version ID** in the same way that decision 122 selected the first **library** to be searched.

Logical search order for version **libraries** is shown in Fig. 6. When a program has a program attribute indicating that version 1 programs are to be run, the search order of the **libraries** is as indicated at 60. The version 1 **library** is first searched. If the command that is searched for is not found in the version 1 **library** 12, the version 2 **library** 14 is searched. Then, if still not found, a first user **library** is searched, then a second. A user **library** is a library which is specified by the user, and usually contains user application programs...

...SPECIFICATION one list or a plurality of lists with each list having a different sequence of **libraries** . The list could be selected based on the version ID.

For example, a command indicated...

...step 132 could use search routine 144 to search libraries according the version priority -- V1 **library** , V"N" **library** , User **library** 1, User **library** 2, etc. This priority or sequence of listed **libraries** could be different for each version and could be selected as a **function** of **version ID** in the same way that decision 122 selected the first **library** to be searched.

Logical search order for version **libraries** is shown in Fig. 6. When a program has a program attribute indicating that version 1 programs are to be run, the search order of the **libraries** is as indicated at 60. The version 1 **library** is first searched. If the command that is searched for is not found in the version 1 **library** 12, the version 2 **library** 14 is searched. Then, if still not found, a first user **library** is searched, then a second. A user **library** is a library which is specified by the user, and usually contains user application programs...

4/3,K/10 (Item 1 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

(c) 2004 WIPO/Univentio. All rts. reserv.

01060367 **Image available**

INITIALIZING, MAINTAINING, UPDATING AND RECOVERING SECURE OPERATION WITHIN AN INTEGRATED SYSTEM EMPLOYING A DATA ACCESS CONTROL FUNCTION
OPERATION SECURISEE D'INITIALISATION, DE MAINTIEN, DE MISE A JOUR ET DE RECUPERATION DANS UN SYSTEME INTEGRE UTILISANT UNE FONCTION DE CONTROLE D'ACCES AUX DONNEES

Patent Applicant/Assignee:

INTERNATIONAL BUSINESS MACHINES CORPORATION, New Orchard Road, Armonk, NJ 10504, US, US (Residence), US (Nationality)

Inventor(s):

FOSTER Eric M, 41 Front Street, Owego, NY 13827, US,
HALL William E, 8 White Oak Drive, Clinton, CT 06413, US,
ROSU Marcel-Catalin, Building 1, Apt. 8, Briarcliff Drive South, Ossining, NY 10562, US,

Legal Representative:

SCHNURMANN Daniel H (agent), International Business Machines Corporation, Dept. 18G/Bldg. 300-482, 2070 Route 52, Hopewell Junction, NY 12533, US

Patent and Priority Information (Country, Number, Date):

Patent: WO 200390402 A1 20031030 (WO 0390402)

Application: WO 2003US11907 20030416 (PCT/WO US0311907)

Priority Application: US 2002125803 20020418

Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ OM PH PL PT RO RU SC SD SE SG SK SL TJ TM TN TR TT TZ UA UG UZ VC VN YU ZA ZM ZW (EP) AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL PT RO SE SI SK TR

(OA) BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG

(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZM ZW

(EA) AM AZ BY KG KZ MD RU TJ TM

Publication Language: English

Filing Language: English

Fulltext Word Count: 9538

International Patent Class: G06F-012/14

Fulltext Availability:

Claims

Claim

... The method of claim 1, in combination with a method for updating a level of **software** , said method for updating the level of software comprising:

determining a software level update is...

...determining the current level of software is authorized to authenticate the updated

25

level of **software**, and if so, proceeding to accept the **software** level update into the integrated system employing the data access control function.

6 The method of claim 5, wherein the updated level of **software** comprises an initialization **code** update, and wherein the method further includes: authenticating and decrypting the initialization **code** update; encrypting the decrypted initialization **code** update with a master key set maintained by the data access control **function** and a new **version number**; and updating an initialization re-direction address maintained by the data access control function to...

...initialization location address.

7 The method of claim 5, wherein if the current level of **software** is unable to authenticate the **software** level update, then the **software** level update is held at the integrated system until an appropriate level of **software** is running and able to authenticate the **software** level update.

8 The method of claim 5, wherein the proceeding to accept the **software** level update into the integrated system comprises encrypting for storage the **software** level update, the encrypting comprising employing a new version number when encrypting the software level...

4/3,K/11 (Item 2 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

(c) 2004 WIPO/Univentio. All rts. reserv.

00989316 **Image available**

METHOD AND SYSTEM FOR FACILITATING CREATION, PRESENTATION, EXCHANGE, AND MANAGEMENT OF DOCUMENTS TO FACILITATE BUSINESS TRANSACTIONS
PROCEDE ET SYSTEME FACILITANT LA CREATION, LA PRESENTATION, LES ECHANGES, ET LA GESTION DE DOCUMENTS, ET PAR LA LES TRANSACTIONS D'AFFAIRES

Patent Applicant/Assignee:

CORE IPR LIMITED, 2606-2601 26/F, China Ressources Bldg., 26 Harbour Road, Wanchai, Hong Kong, HK, CN (Residence), -- (Nationality)

Inventor(s):

HUNG Michael, 2606-2601, 26F, China Ressources Bldg., 26 Harbour Road, Wanchai, Hong Kong, CN,

LAM Marcus, 2606-2601, 26/F, China Ressources Bldg., 26 Harbour Road, Wanchai, Hong Kong, CN,

TAI Patrick, 2606-2601, 26/F, China Ressources Bldg., 26 Harbour Road, Wanchai, Hong Kong, CN,

Legal Representative:

DINH Truong T (agent), Dinh & Associates, 2506 Ash Street, Palo Alto, CA 94306, US,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200319326 A2-A3 20030306 (WO 0319326)

Application: WO 2002US14916 20020509 (PCT/WO US0214916)

Priority Application: US 2001290079 20010509

Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU

CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP

KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ OM PH PL PT RO

RU SD SE SG SI SK SL TJ TM TN TR TT TZ UA UG UZ VN YU ZA ZM ZW

(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR

(OA) BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG

(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZM ZW

(EA) AM AZ BY KG KZ MD RU TJ TM

Publication Language: English

Filing Language: English

Fulltext Word Count: 12806

Main International Patent Class: G06F-017/60

International Patent Class: G06F-007/00

Fulltext Availability:

Detailed Description

Detailed Description

... system and the organizations registered with the ASP, and farther leverages on the existing facilities. **Application** programming interface (API) processes may be used for this purpose, which enable the exchange of to the document by the sender using an amendment **function** ..

When a document is amended, its **version number** is automatically incremented, indicating the latest version of the document. By default, when an amendment...

4/3,K/12 (Item 3 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

(c) 2004 WIPO/Univentio. All rts. reserv.

00963611 **Image available**

EXTENDED WEB ENABLED MULTI-FEATURED BUSINESS TO BUSINESS COMPUTER SYSTEM FOR RENTAL VEHICLE SERVICES
SYSTEME INFORMATIQUE INTERENTREPRISES A ELEMENTS MULTIPLES A ACCES INTERNET POUR SERVICES DE LOCATION DE VEHICULES

Patent Applicant/Assignee:

THE CRAWFORD GROUP INC, 600 Corporate Park Drive, St. Louis, MO 63105, US
, US (Residence), US (Nationality), (For all designated states except: US)

Patent Applicant/Inventor:

WEINSTOCK Timothy Robert, 1845 Highcrest Drive, St. Charles, MO 63303, US
, US (Residence), US (Nationality), (Designated only for: US)
DE VALLANCE Kimberly Ann, 2037 Silent Spring Drive, Maryland Heights, MO 63043, US, US (Residence), US (Nationality), (Designated only for: US)
HASELHORST Randall Allan, 1016 Scenic Oats Court, Imperial, MO 63052, US, US (Residence), US (Nationality), (Designated only for: US)
KENNEDY Craig Stephen, 9129 Meadowglen Lane, St. Louis, MO 63126, US, US (Residence), US (Nationality), (Designated only for: US)
SMITH David Gary, 10 Venice Place Court, Wildwood, MO 63040, US, US (Residence), US (Nationality), (Designated only for: US)
TINGLE William T, 17368 Hilltop Ridge Drive, Eureka, MO 63025, US, US (Residence), US (Nationality), (Designated only for: US)
KLOPFENSTEIN Anita K, 433 Schwarz Road, O'Fallon, IL 62269, US, US (Residence), US (Nationality), (Designated only for: US)

Legal Representative:

HAFERKAMP Richard E (et al) (agent), Howell & Haferkamp, L.C., Suite 1400, 7733 Forsyth Blvd., St. Louis, MO 63105-1817, US,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200297700 A2 20021205 (WO 0297700)

Application: WO 2001US51431 20011019 (PCT/WO US0151431)

Priority Application: US 2000694050 20001020

Parent Application/Grant:

Related by Continuation to: US 2000694050 20001020 (CIP)

Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PH PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG US UZ VN YU ZA ZW
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR
(OA) BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW
(EA) AM AZ BY KG KZ MD RU TJ TM

Publication Language: English

Filing Language: English

Fulltext Word Count: 237932

Main International Patent Class: G06F-017/60

Fulltext Availability:
Detailed Description

Detailed Description

... Age, Driver's License 9, State, and Expiration Date.

3. Key Y=Yes, or N= No , if the rental will be ...Key "Bill To"
Information: Company Name, Contact Person, Address, Phone Number, City,
State, and Zip Code will be preloaded, if valid Customer # is keyed on
the first Reservation Screen.

S. Pickup...

4/3,K/13 (Item 4 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

(c) 2004 WIPO/Univentio. All rts. reserv.

00865357 **Image available**

SYSTEM FOR INTELLECTUAL PROPERTY REUSE IN INTEGRATED CIRCUIT DESIGN
SYSTEME DE REUTILISATION D'ELEMENTS DE PROPRIETE INTELLECTUELLE EN
CONCEPTION DE CIRCUIT INTEGRE

Patent Applicant/Assignee:

MENTOR GRAPHICS CORP, 8005 SW Boeckman Drive, Wilsonville, OR 97070-7777,
US, US (Residence), US (Nationality), (For all designated states
except: US)

Patent Applicant/Inventor:

GARNER Robert E, 3456 Mulberry Creek Drive, Austin, TX 78732, US, US
(Residence), US (Nationality), (Designated only for: US)

GARDNER David N, 2128 Woodston Drive, Round Rock, TX 78681, US, US
(Residence), US (Nationality), (Designated only for: US)

JUSSEL Jeffrey, 9564 Indigo Brush Drive, Austin, TX 78726, US, US
(Residence), US (Nationality), (Designated only for: US)

ELMAN Anna, 10703 Cassia Drive, Austin, TX 78759, US, US (Residence), US
(Nationality), (Designated only for: US)

LING David, 12714 Red Deer Pass, Austin, TX 78729-6436, US, US
(Residence), US (Nationality), (Designated only for: US)

BENAVIDES Alvaro Eduardo, 1306 Creekview Drive, Round Rock, TX 78681, US,
US (Residence), US (Nationality), (Designated only for: US)

MCADAMS Mark A, 5508 Lomita Circle, Plano, TX 75023, US, US (Residence),
US (Nationality), (Designated only for: US)

Legal Representative:

SCHEINBERG Michael O (agent), P.O. Box 164140, Austin, TX 78716-4140, US,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200198921 A1 20011227 (WO 0198921)

Application: WO 2001US20131 20010621 (PCT/WO US0120131)

Priority Application: US 2000213207 20000621

Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ

DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ

LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG

SI SK SL TJ TM TR TT TZ UA UG US UZ VN YU ZA ZW

(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR

(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG

(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW

(EA) AM AZ BY KG KZ MD RU TJ TM

Publication Language: English

Filing Language: English

Fulltext Word Count: 9510

Main International Patent Class: G06F-015/18

Fulltext Availability:

Claims

Claim

... the component; and

a deliverable version including one or more design files performing

a common function and having described by deliverable version number

2 The system of claim 1 further comprising metadata associating the deliverable version number with...

...which the deliverable files are stored.

4 The system of claim 3 further comprising an **application** server through which the user accesses the database to create and modify the metadata database...

4/3,K/14 (Item 5 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00859411 **Image available**

A SOFTWARE DEVELOPMENT SYSTEM THAT PRESENTS A LOGICAL VIEW OF PROJECT COMPONENTS, FACILITATES THEIR SELECTION, AND SIGNALS MISSING LINKS PRIOR TO COMPILATION

SYSTEME LOGICIEL DE MISE AU POINT PRESENTANT UNE VUE LOGIQUE DE COMPOSANTS PROJET, FACILITANT LEUR SELECTION, ET LIAISONS MANQUANTES DE SIGNAUX AVANT LA COMPILATION

Patent Applicant/Assignee:

PHOENIX TECHNOLOGIES LTD, 411 E. Plumeria Drive, San Jose, CA 95134, US,
US (Residence), US (Nationality)

Inventor(s):

COHEN Frances, 65 Carson, Irvine, CA 92620, US,
BOMBET Marc Abraham, 22501 Chase #8103, Aliso Viejo, CA 92656, US,
LUSINSKY Robert Dennis, 1449 E. Holcomb Place, Placentia, CA 92870, US,
LEWIS Timothy Andrew, 33775 Shallow Court, Fremont, CA 94555, US,
SANDUSKY Marc Shane, 7 Greenborrow Court, Aliso Viejo, CA 92565, US,

Legal Representative:

NOBLES Kimberley G (agent), Irell & Manella, LLP, Suite 400, 840 Newport Center Drive, Newport Beach, CA 92660, US,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200193031 A1 20011206 (WO 0193031)
Application: WO 2001US9094 20010320 (PCT/WO US0109094)
Priority Application: US 2000531678 20000320

Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU

CZ DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR
KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE
SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR
(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW
(EA) AM AZ BY KG KZ MD RU TJ TM

Publication Language: English

Filing Language: English

Fulltext Word Count: 60485

Main International Patent Class: **G06F-009/45**

Fulltext Availability:

Detailed Description

Detailed Description

... the version number of the function could be encoded as a

-style keyword in the **function** prototype.

In addition, the exact manner by which the **version number** is attached to the caller and callee could vary from programming language to **programming language**. For example, in C, it can be attached to the function prototypes if MASM macros...

...can also apply to static variables or data structures.

This use of versioning allows large **software** projects to be built more independently by tracking semantic changes in input parameters (bug

fixes/modified parameter definitions). It also gives improved feedback as to whether the **software** project is likely to work before compilation and run-time. The versioning may also reduce...

4/3,K/15 (Item 6 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

(c) 2004 WIPO/Univentio. All rts. reserv.

00811354 **Image available**

A METHOD FOR TRACING ERRORS IN A SOFTWARE

PROCEDE DE DEPISTAGE DES ERREURS DANS UN LOGICIEL

Patent Applicant/Assignee:

TELEFONAKTIEBOLAGET LM ERICSSON (publ), S-126 25 Stockholm, SE, SE
(Residence), SE (Nationality)

Inventor(s):

BODO Bela, Kanelgrand 44, S-135 36 Tyreso, SE,

Legal Representative:

NORIN Klas (agent), Ericsson Radio Systems AB, Common Patent Department,
S-164 80 Stockholm, SE,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200144931 A1 20010621 (WO 0144931)

Application: WO 2000SE2558 20001215 (PCT/WO SE0002558)

Priority Application: SE 994646 19991217

Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ

DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ

LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG

SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW

(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR

((OAPI utility model)) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG

(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW

(EA) AM AZ BY KG KZ MD RU TJ TM

Publication Language: English

Filing Language: English

Fulltext Word Count: 3002

Main International Patent Class: **G06F-009/44**

Fulltext Availability:

Claims

Claim

... method of facilitating the tracing of errors in function library software, wherein said function library **software** is achieved in a building process, i.e. the compiling and linking of a number of source- **code** files which are stored in a version controlling system, and wherein at least a part...

...said building process

furthermore results in a record which specifies names and versions of source- **code** files included in said building process c h a r a c t e r...and version number of the record thus stored, and

-bundling (206, 207) said path and **version number** with said **function library software** in such a manner that said path and **version number** is retrievable when the **function library software** is to be used.

6 A computer readable medium comprising, in compiled format, the following...

4/3,K/16 (Item 7 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

(c) 2004 WIPO/Univentio. All rts. reserv.

00790513 **Image available**

SOFTWARE DEVELOPMENT SYSTEM FOR FACILITATING SELECTION OF COMPONENTS
SYSTEME LOGICIEL DE MISE AU POINT POUR FACILITER LA SELECTION DE COMPOSANTS

Patent Applicant/Assignee:

PHOENIX TECHNOLOGIES LTD, 411 East Plumeria Drive, San Jose, CA 95134, US
, US (Residence), US (Nationality)

Inventor(s):

COHEN Frances, 65 Carson, Irvine, CA 92620, US,
BOMBET Marc Abraham, 22501 Chase #8103, Aliso Viejo, CA 92656, US,
LUSINSKY Robert Dennis, 1449 E. Holcomb Place, Placentia, CA 92870, US,
LEWIS Timothy Andrew, 33775 Shallow Ct., Fremont, CA 94555, US,
SANDUSKY Marc Shane, 7 Greenborrow Court, Aliso Viejo, CA 92656, US,

Legal Representative:

RITTMASER Ted R (agent), Foley & Lardner, Suite 3500, 2029 Century Park
East, Los Angeles, CA 90067-3021, US,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200123998 A1 20010405 (WO 0123998)
Application: WO 2000US25928 20000922 (PCT/WO US0025928)
Priority Application: US 99404298 19990924; US 2000531678 20000320

Designated States: CN DE GB JP

Publication Language: English

Filing Language: English

Fulltext Word Count: 49598

Main International Patent Class: G06F-009/44

Fulltext Availability:

Detailed Description

Detailed Description

... version.

The PUBEXT macro is used to declare a reference to a public interface, either **function** or array. One of the parameters includes the **version number** expected by the caller. The version number has a similar format to that of the...

...database update) utility, which, when invoked, scans for the previously described macros in all source **code** files in each object and places the name of the interface and its version in...

...handled by a preprocessor or special version of a compiler. For example, in C, the **version number** of the **function** could be encoded as a -style keyword in the **function** prototype.

In addition, the exact manner by which the **version number** is attached to the caller and callee could vary from **programming language** to **programming language**.

For example, in C, it can be attached to the function prototypes if MASM macros...

...can also apply to static variables or data structures.

This use of versioning allows large **software** projects to be built more independently by tracking semantic changes in input parameters (bug fixes-modified parameter definitions). It also gives improved feedback as to whether the **software** project is likely to work before compilation and run-time. The versioning may also reduce...

4/3,K/17 (Item 8 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

(c) 2004 WIPO/Univentio. All rts. reserv.

00498874 **Image available**

APPARATUS AND METHOD FOR ALLOWING OBJECT-ORIENTED PROGRAMS CREATED WITH
DIFFERENT FRAMEWORK VERSIONS TO COMMUNICATE

**DISPOSITIF ET PROCEDE PERMETTANT A DES PROGRAMMES ORIENTES OBJET, CREES
AVEC DIFFERENTES VERSIONS DE SYSTEMES, DE COMMUNIQUER**

Patent Applicant/Assignee:

OBJECT TECHNOLOGY LICENSING CORPORATION,

Inventor(s):

JABLONSKI Marc,

DAVIS Mark,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9930226 A2 19990617

Application: WO 98US25993 19981208 (PCT/WO US9825993)

Priority Application: US 97986996 19971208

Designated States: CA JP AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT
SE

Publication Language: English

Fulltext Word Count: 14500

Main International Patent Class: **G06F-009/44**

Fulltext Availability:

Detailed Description

Detailed Description

... and modifications made in accordance with the invention, are discussed
in detail below. The illustrative **code** fragment is as follows.

```
TStream& C::operator>>=(TStream* toWhere)
```

```
WriteVersion(toWhere);
```

```
A::operator>>=(toWhere);
```

```
B...
```

...operator for resurrecting the object

flattened by the above procedure is illustrated below. The ReadVersion(
function is the counterpart of the WriteVersion(**function** which reads
the **version number** . It is also discussed in detail below.

```
TStream& C::operator<<=(TStream* fromWhere)
```

```
VersionInfo v = ReadVersion...
```

...Hread the old version

```
...
```

```
I
```

```
return fromWhere;
```

Overview

The overall issue of data compatibility between **application** programs
constructed with different framework versions encompasses several sub
issues. For example, data incompatibility can...

...when incompatibilities are processed. Finally, there are issues
concerning the reading of data streams by **applications** which were
constructed without using any of the underlying framework versions.

1 . Mismatched class versions...

4/3,K/18 (Item 9 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

(c) 2004 WIPO/Univentio. All rts. reserv.

00332974 **Image available**

SECURITY APPARATUS AND METHOD

DISPOSITIF ET PROCEDE RELATIFS A LA SECURITE

Patent Applicant/Assignee:

ABSOLUTE SOFTWARE CORPORATION,

COTICHINI Christian,

CAIN Fraser,

NGUYEN Thanh Cam,

Inventor(s):

COTICHINI Christian,

CAIN Fraser,

NGUYEN Thanh Cam,
Patent and Priority Information (Country, Number, Date):
Patent: WO 9615485 A1 19960523
Application: WO 95CA646 19951115 (PCT/WO CA9500646)
Priority Application: US 94339978 19941115
Designated States: AL AM AT AU BB BG BR BY CA CH CN CZ DE DK EE ES FI GB GE
HU IS JP KE KG KP KR KZ LK LR LT LU LV MD MG MK MN MW MX NO NZ PL PT RO
RU SD SE SG SI SK TJ TM TT UA UG US UZ VN KE LS MW SD SZ UG AT BE CH DE
DK ES FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN ML MR NE SN
TD TG
Publication Language: Spanish
Fulltext Word Count: 44482

Main International Patent Class: G06F-001/00
Fulltext Availability:
Detailed Description

Detailed Description
... include "SNTLAPI.INC"
include "SNTLDATA.INC"
include "SNTLTIMR.INC"
%LIST
SEGMENT SNTL SEG BYTE PUBLIC ' CODE '

;* SNTLAPI
;* PURPOSE:
;* This function provides an external API for the Ward and Tender
modules,
;* as well as development software tools, to gain access to the
Sentinel

;* The following functions are supported:
;* Function ...1 - Set Sentinel State to ALERT
;* returns CF = 0 if successful
;* CF = 1 if failed
;* Function 2 - Get Sentinel version Number
;* returns AH = major version number
;* AL = minor version number
;* Function 3 - Get Sentinel Serial Number
;* returns ES:DI = pointer to serial number
;* Function 4 - Cancel...

4/3,K/19 (Item 10 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00283451
SYSTEM FOR OBJECT ORIENTED DYNAMIC LINKING BASED UPON A CATALOG OF
REGISTERED FUNCTION SET OR CLASS IDENTIFIERS
SYSTEME DE LIAISON DYNAMIQUE ORIENTEE OBJETS SUR LA BASE D'UN CATALOGUE
D'IDENTIFICATEURS ENREGISTRES DE JEUX OU DE CLASSES DE FONCTIONS

Patent Applicant/Assignee:

APPLE COMPUTER INC,

Inventor(s):

COWSAR George Curtis,
PLUMMER Christopher Jordan,
QUINN Michael John,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9501598 A1 19950112
Application: WO 94US7424 19940630 (PCT/WO US9407424)
Priority Application: US 93187 19930630

Designated States: AT AU BB BG BR BY CA CH CN CZ DE DK ES FI GB GE HU JP KE
KG KP KR KZ LK LU LV MD MG MN MW NL NO NZ PL PT RO RU SD SE SI SK TJ TT
UA UZ VN AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI
CM GA GN ML MR NE SN TD TG

Publication Language: English
Fulltext Word Count: 40073

Main International Patent Class: G06F-009/44
International Patent Class: G06F-09:445
Fulltext Availability:
Detailed Description
Claims

Detailed Description

... number fields are used to insure that the cached information is still valid (if the **library** unloaded and a new version of a class or function set in the **library** was dragged into a registered folder, any cached information is invalid). The version number fields contain information about the **version number** of the class or **function** set that your **application** (**code** resource, extension, etc.) linked with, and the f ClassIDStr field contains the actual ClassID of...kind of objectoriented ability for ordinary functions.

version

This declaration defines the version of the **function** set. The **version number**

is in the standard Apple **version number** form: The version number may not have the extra release information (like b2) on it...

...numbers separated either by 3 dots (...) or an ellipsis character (option-;). This indicates the minimum **version number** of the **function** set that this **function** set is backwards-compatible with, and the current **version number** of the **function** set. Nothing is done with this information in version 1.0 of SLM, but future...

...information. If you do not specify a version number, the version number specified in the " **Library** " declaration will be assumed. This may be a #defined symbol.

exports =

This declares a comma...

Claim

... link cache in response to the return of the particular member function to the client **application** ;
and
the dispatch **code** includes a routine which looks in the set link cache for a cached link to...

...to a standard protocol, and dispatch record further includes version information linked with the client **application** indicating a minimum version number supported by the client **application** for the **function** set of the particular member **function** ,
the **library** includes a **version number** for the corresponding **function** sets,
and
the link routine includes **code** responsive to the version information in the dispatch record and the version number in the **library** . to insure that the client **application** supports a version of the function set of the particular member function. I 11. The apparatus of claim 9, wherein the function sets in the **library** are assigned serial numbers when the function sets are loaded in internal memory, and the dispatch record further includes a serial number linked with the client **application** indicating a serial number of the corresponding function set when the set link cache is...non-virtual function.

31 The apparatus of claim 26, wherein the function sets in the **library** are assigned version numbers according to a standard protocol, and dispatch record further includes version information linked with the client indicating a minimum **version number** supported by the client for the **function** set of which the particular **function** is a member,
the set record includes a **version number** for the corresponding **function** set,
and

the link engine includes a routine responsive to the version information in the...numbers according to a standard protocol, and including version information linked 163 with the client **application** indicating a minimum version number supported by the client **application** for the **function** set of the particular member **function**, the **library** includes a **version number** for the corresponding **function** sets, and the link routine includes **code** responsive to the version information in the dispatch record and the version number in the **library** to insure that the client **application** supports a version of the function set of the particular member function.

52 An apparatus for managing **code** resources for use by client **applications** in a computer, the computer having internal memory storing at least one client **application**, comprising:
a class identifier and link pointer in the client **application**, upon compiling the client **application**;
a class catalog, the catalog stored in the internal memory, the class catalog identifying a...

4/3,K/20 (Item 11 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00279043

METHOD AND APPARATUS FOR VECTORIZING THE CONTENTS OF A READ ONLY MEMORY DEVICE WITHOUT MODIFYING UNDERLYING SOURCE CODE
PROCEDE ET APPAREIL POUR VECTORISER LE CONTENU D'UN DISPOSITIF A MEMOIRE ROM SANS MODIFIER LE CODE SOURCE SOUS-JACENT

Patent Applicant/Assignee:

APPLE COMPUTER INC,

Inventor(s):

WETMORE Russ,
NGUYEN Philip,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9427220 A1 19941124

Application: WO 94US4994 19940506 (PCT/WO US9404994)

Priority Application: US 9358876 19930506

Designated States: AT AU BB BG BR BY CA CH CN CZ DE DK ES FI GB GE HU JP KG
KP KR KZ LK LU LV MD MG MN MW NL NO NZ PL PT RO RU SD SE SI SK TJ TT UA
UZ VN AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM
GA GN ML MR NE SN TD TG

Publication Language: English

Fulltext Word Count: 7124

Main International Patent Class: **G06F-009/45**

Fulltext Availability:

Detailed Description

Detailed Description

... vectorization tool. It is then recompiled using an alternate set of macros to produce the **code** that initializes the vector table.

Patchina A Vectorized ROM

As the purpose of vectorizing the **code** is to facilitate fixing bugs or adding functionality, it is now useful to describe how..

...patches results from the need to support prior versions of the ROM and the ROM **code** with each released ROM the vectorized routine will have a **version number**. If a bug is discovered or a new **function** is added to the routine, then the new routine will have a higher version number...

4/3,K/21 (Item 12 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00279042

**METHOD AND APPARATUS FOR PATCHING CODE RESIDING ON A READ ONLY MOEMORY
DEVICE**

**PROCEDE ET APPAREIL POUR LA CORRECTION DE CODES RESIDANT DANS UN DISPOSITIF
A MEMOIRE ROM**

Patent Applicant/Assignee:

APPLE COMPUTER INC,

Inventor(s):

WETMORE Russ,

NGUYEN Philip,

BATISTA Ricardo,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9427219 A1 19941124

Application: WO 94US4995 19940506 (PCT/WO US9404995)

Priority Application: US 9358877 19930506

Designated States: AT AU BB BG BR BY CA CH CN CZ DE DK ES FI GB GE HU JP KG

KP KR KZ LK LU LV MD MG MN MW NL NO NZ PL PT RO RU SD SE SI SK TJ TT UA

UZ VN AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM

GA GN ML MR NE SN TD TG

Publication Language: English

Fulltext Word Count: 7701

Main International Patent Class: **G06F-009/44**

Fulltext Availability:

Detailed Description

Detailed Description

... that initializes the vector table.

Patchina A Vectorized ROM

As the purpose of vectorizing the **code** is ...patches results from the
need to support

prior versions of the ROM and the ROM **code** with each released ROM the
vectorized routine will have a **version number** . If a bug is discovered
or a

new **function** is added to the routine, then the new routine will have a
higher version number...

4/3,K/22 (Item 13 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00272734

DISTRIBUTED WORK FLOW MANAGEMENT

GESTION REPARTIE DU DEROULEMENT DES OPERATIONS

Patent Applicant/Assignee:

NAUCKHOFF Sven,

Inventor(s):

NAUCKHOFF Sven,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9420910 A1 19940915

Application: WO 94SE173 19940301 (PCT/WO SE9400173)

Priority Application: SE 93671 19930301

Designated States: US AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE

Publication Language: English

Fulltext Word Count: 27270

Main International Patent Class: **G06F-015/21**

Fulltext Availability:

Detailed Description

Detailed Description

... SHEET

Some implementation notes.

1. The DEACTIVATE CASE is performed by the ATX when the
Application Task has terminated.

If the case has been created or changed, then the
function stores the new case **version number** in the case If the
current case is marked for deletion (see
TERMINATE-CASE above...

4/3,K/23 (Item 14 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00183685 **Image available**
A METHOD AND APPARATUS FOR INFORMATION MANAGEMENT IN A COMPUTER SYSTEM
PROCEDE ET APPAREIL DE GESTION D'INFORMATIONS DANS UN SYSTEME INFORMATIQUE
Patent Applicant/Assignee:
POQET COMPUTER CORPORATION,
Inventor(s):
CULLIMORE Ian H S,
Patent and Priority Information (Country, Number, Date):
Patent: WO 9101024 A2 19910124
Application: WO 90US3680 19900628 (PCT/WO US9003680)
Priority Application: US 89691 19890630
Designated States: AT AT AU BB BE BF BG BJ BR CA CF CG CH CH CM DE DE DK DK
ES ES FI FR GA GB GB HU IT JP KP KR LK LU LU MC MG ML MR MW NL NL NO RO
SD SE SE SN SU TD TG
Publication Language: English
Fulltext Word Count: 20163

Main International Patent Class: **G06F-012/00**

Fulltext Availability:

Detailed Description

Detailed Description

... the range

1 - FFh)

Function 5:MaDjUrmap Handle Pages

Passed: AR 44h Returns: AR Error **code**

AL Mysical Page

Number

BX Logical Page Number

DX RMN Handle

if E8:DI points assumed

Function 6:Deallocate Pages

15 Passed: AR 49h Returns: AR Error **code**

DX RMN Ran e

Function 7:Get Version

Passed: AR 46h Returns: AR Error **code**

AL version number

AL is returned as

major/minor

BCD **version**

number , e.g.